Predicting Health-related Quality of Life in Early Adulthood from Individual Differences in Adolescent Neurocognitive Development

by

Jennifer M. Fedor

BS, Neuroscience and Psychology, University of Pittsburgh, 2014

Submitted to the Graduate Faculty of

the Department of Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

Jennifer M. Fedor

It was defended on

April 20, 2020

and approved by

Thesis Advisor:

Hanna Bandos, PhD, Research Assistant Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Committee Members:

Jeanine Buchanich, MPH, MEd, PhD, Research Associate Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Jenna Carlson, PhD, Assistant Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Beatriz Luna, PhD, Professor, Departments of Psychiatry and Psychology, School of Medicine and Kenneth P. Dietrich School of Arts and Sciences, University of Pittsburgh

Ada Youk, PhD, Associate Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Hanna Bandos, PhD

Predicting Health-related Quality of Life in Early Adulthood from Individual Differences in Adolescent Neurocognitive Development

Jennifer M. Fedor, MS

University of Pittsburgh, 2020

Abstract

Background: Inhibitory control and the brain systems that support this cognitive process are known to undergo a protracted maturation through adolescence. Relatively little work, however, has examined how individual differences in the normative adolescent development of inhibitory control may be associated with outcomes such as health-related quality of life (QOL) in early adulthood.

Methods: We analyzed data from an accelerated longitudinal study of healthy individuals initially aged 8-30 who, at approximately yearly intervals, completed an inhibitory control task while functional magnetic resonance imaging data were acquired. Generalized additive mixed models were utilized to characterize age-related change in inhibitory control behaviorally and in regional brain activation. Random intercepts and slopes from these models, representing person-specific deviations from group-level developmental trajectories, were utilized as covariates in a bootstrap-enhanced elastic net regression procedure to predict QOL, which was assessed with a self-report questionnaire at the study endpoint.

Results: There were significant developmental improvements in inhibitory control behaviorally that continued into young adulthood. Among examined motor response and executive control brain regions, there were significant age-related decreases in activation during correctly performed task trials occurring until mid-adolescence in the L frontal eye fields (FEF), bilateral posterior parietal cortex (pPC), and R dorsolateral prefrontal cortex (dlPFC). In the performance monitoring region, dorsal anterior cingulate (dACC), activation during error-corrected trials significantly increased with age and reached mature levels in young adulthood. The bootstrap-enhanced elastic net model indicated that person-specific deviations from these adolescent developmental trajectories did not significantly predict QOL in early adulthood, although variable inclusion probabilities suggested

that performance monitoring behaviorally and activation in L FEF and R dlPFC may be relatively important predictors.

Conclusions: Findings show that subtle age-related changes in inhibitory control may continue later into adolescence and young adulthood than previously reported. Individual differences in adolescent development of aspects of inhibitory control may potentially be important predictors of QOL in early adulthood, but require further investigation.

Public health significance: Establishing the relationship between individual trajectories of neurocognitive maturation and subsequent QOL may help to inform the development of personalized interventions that can be applied during adolescence to promote optimal adult outcomes.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

Adolescence is a unique period of development during which many social, biological, and cognitive changes take place. In particular, significant changes in brain structure and function, such as synaptic pruning and myelination [1] and the functional integration of distributed circuitry [2], occur during this period. These changes in brain structure and function during adolescence are thought to support continued refinements in cognitive control, the ability to voluntarily and adaptively coordinate goal-directed behavior [3]. Cognitive control is an essential process, as it is necessary for the performance of more complex cognitive tasks such as planning, reasoning, and decision-making [4]; is involved in emotion regulation [5]; and plays a significant role in health and health behavior [6].

Cognitive control is of particular interest during adolescence because of its relationships with risk-taking behavior and psychopathology, which increase and emerge, respectively, during this stage of development. More specifically, immaturities in cognitive control that persist into adolescence, coupled with concomitant increases in sensation-seeking and sensitivity to rewards, may make adolescents particularly vulnerable to risk-taking and sub-optimal decision-making [7]. This is evidenced by transient increases in mortality due to preventable injury, including motor vehicle crashes and suicide, during this period [8, 9]. Additionally, psychiatric disorders, many of which are characterized by deficits in cognitive control [10], and substance use are significant contributors to morbidity in adolescence [11], and can have lasting effects on health into adulthood.

Improving the health and well-being of adolescents and young adults, and supporting healthy development during the transition from adolescence to young adulthood, have been identified as important public health goals [12]. Prior research has been instrumental in characterizing the normative development of cognitive control and underlying brain regions during adolescence. However, adolescents are a heterogeneous group; relatively little work has explored individual differences in adolescent developmental trajectories, and how such individual differences may contribute to outcomes such as health-related quality of life in early adulthood. As a subjective measure of health and well-being, quality of life can help to identify hidden morbidities that would otherwise go undetected, and thus may be a particularly informative outcome measure. Because adolescence is a time of significant plasticity, it is amenable to and thus a promising target for public health intervention. Therefore, establishing the relationship between individual trajectories of neurocog-

nitive maturation in adolescence and health-related quality of life in adulthood may help to inform the development of personalized interventions that can be applied during adolescence in order to ultimately promote optimal adult outcomes.

1.1 Inhibitory control

Cognitive control is comprised of distinct, but overlapping, component processes including working memory, task-switching, and inhibitory control [13–15]. Inhibitory control, sometimes referred to as (voluntary) response inhibition, describes the ability to inhibit a reflexive or prepotent, goal-irrelevant response in favor of a goal-appropriate response [4]. Inhibitory control is present early in development but undergoes a protracted maturation during adolescence.

Inhibitory control is evident as early as infancy, albeit in a basic form. Infants are able to inhibit prepotent reaching responses [16] and suppress attention to distractor stimuli [17]. As inhibitory control begins to improve in early childhood [18] and continues into adolescence [19, 20], individuals are able to perform more complex inhibitory tasks, although the age at which performance reaches adult-like levels may depend on the specific task being examined. Popular paradigms for measuring inhibitory control include the Stroop task, in which individuals must respond with the color in which a word is printed, which on some trials is incongruent with the word itself (e.g., "red" printed in green); the stop-signal task, in which individuals must suppress an already-initiated response such as a button press when an unpredictable cue is presented; and the antisaccade task [3, 4].

The antisaccade task [21] is a well-established oculomotor task that has been widely used to study age-related changes in inhibitory control. In the antisaccade task, participants are presented with a peripheral stimulus, such as a flash of light. In order to perform the task correctly, participants must inhibit the prepotent response to make a saccade to the stimulus location (i.e., a prosaccade), and instead make a saccade to the mirror-image location (i.e., an antisaccade). Young children perform relatively poorly on the antisaccade task. Compared to older age groups, children aged 5 to 8 have more variable and slower overall saccade latencies, and make more errors due to difficulty in suppressing prepotent prosaccades [19]. Extending these findings, in a cross-sectional sample of individuals aged 8 to 30, Luna and colleagues [20] reported that antisaccade latency and the proportion of prosaccade-suppression errors declined rapidly from late childhood through adolescence, following an inverse curve trajectory. Change-point analysis revealed that adult levels of

antisaccade performance were reached at age 14. These results provide evidence that the capacity for inhibitory control is available in childhood, as all children in the study were able to correctly perform an antisaccade on at least one task trial, but the ability to instantiate inhibitory control consistently improves during adolescence. This suggests that improvement in inhibitory control during adolescence reflects refinement of an extant cognitive ability, rather than the acquisition of a fundamentally new process [3, 20]. These findings were replicated in a recent longitudinal study, which also found that antisaccade latencies and error rates decelerated from late childhood into young adulthood [22].

1.1.1 Brain systems supporting inhibitory control

Improvements in inhibitory control during adolescence are supported by concurrent changes in regional brain activity. Functional magnetic resonance imaging (fMRI) studies have been instrumental in advancing our understanding of the neural substrates that support the transition to mature levels of inhibitory control during adolescence. Blood oxygenation level dependent (BOLD) fMRI is a non-invasive technique that detects regional changes in blood flow and oxygenation that occur when neuronal activity is upregulated in response to a cognitive task; thus, fMRI provides an indirect measure of brain activity [23].

Traditionally, fMRI studies investigating the development of the neural substrates of inhibitory control have focused on the prefrontal cortex (PFC), a region that has an integral role in the top-down control of behavior [14, 24] and is known to undergo a protracted maturation structurally [25]. Results of these studies have been inconsistent, with both age-related increases [2, 26, 27] and decreases [28, 29] in inhibitory task-related activity in PFC regions being reported. Providing more compelling evidence for the latter finding, a recent longitudinal study investigating the functional development of brain systems supporting antisaccade performance reported that activity in the right dorsolateral PFC (dlPFC) significantly decreased with age through adolescence, although this was not associated with improvements in task performance [22]. Developmental decreases in PFC activation have been interpreted as potentially reflecting reduced effort required for implementing inhibitory control in adulthood [30]. Such decreases have also been posited to reflect developmental increases in the functional integration of a number of brain regions that support inhibitory control, which consequently reduces demands on these regions individually [31, 32].

Indeed, inhibitory control is supported by a distributed circuitry comprised of key executive regions, including the dlPFC and ventrolateral prefrontal cortex (vlPFC); motor regions such as the supplementary motor area, posterior parietal cortex, and putamen; and the dorsal anterior cingulate (dACC) [2, 33]. Motor regions are essential for planning, preparing, and executing motor responses, such as eye movements in the antisaccade task, but typically do not exhibit age-related changes in activation from late childhood into adolescence or adulthood [22, 34]. This suggests that motor response systems necessary for inhibitory control are in place early in development, consistent with evidence that, structurally, sensorimotor areas mature relatively earlier than do higher-order association areas including the PFC [25]. The dACC contributes to the development of inhibitory control due to its role in performance monitoring and error processing [35]. Activation of the dACC has been found to increase with age on error-corrected antisaccade trials, where a saccade to the correct location follows an initial failure to inhibit the prepotent prosaccade, and activity in this region reaches mature levels later than does the dlPFC [22, 34]. Further, dACC activation was found to significantly mediate the relationship between antisaccade performance and age [22], suggesting that dACC activity uniquely supports developmental improvements in inhibitory control observed behaviorally.

In sum, the literature suggests that behavioral refinements in inhibitory control during adolescence are supported by the transition from primary reliance on prefrontal executive systems, to reliance on a circuitry comprised of prefrontal executive, sensorimotor, and attentional systems. Specifically, age-related decreases in PFC activation may reflect reductions in the amount of effort involved in reliably instantiating inhibitory control from childhood through adolescence, as well as the integration and recruitment of additional brain regions to support inhibitory control processes. One such additional area that appears to be of particular importance is the dACC due to its function in performance monitoring. Altogether, this evidence is consistent with changes in brain structure known to occur during adolescence (e.g., myelination), which are thought to support efficient processing locally as well as promote connections between distributed systems.

1.2 Health-related quality of life

Health-related quality of life refers to an individual's subjective perception of their health and well-being; it is a multidimensional concept that encompasses physical, psychological, and social

domains [36]. This conceptualization dovetails with the World Health Organization's (WHO) definition of health as a "state of complete physical, mental and social well-being and not merely the absence of disease" [37]. Much of the initial work investigating determinants of health-related quality of life focused on contextual factors, such as sociodemographic characteristics. A number of sociodemographic factors including male gender [38, 39], high socioeconomic status and subjective social class [40], and high level of educational attainment [40, 41] have been linked to better health-related quality of life, overall as well as in specific domains. More recently, however, there has been a focus on the contribution of intrapersonal factors, including affect and cognition, which evidence suggests are stronger or more long-term predictors of subjective well-being than are contextual factors [42].

Previous research explicitly investigating the relationship between inhibitory control and health-related quality of life has been sparse and has yielded conflicting results. One study of older men and women with cardiovascular disease found that inhibitory control, assessed using a modified version of the Stroop task, was significantly related to a measure of independent functioning in daily activities [43]. Another study of older but otherwise healthy women aged 65 to 75 found no significant association between inhibitory control, as measured by performance on the Stroop task, and quality-adjusted life years, although other components of cognitive control such as working memory were found to be significantly related to this measure of health-related quality of life [44]. The inconsistencies between these findings may potentially be explained by differences in the populations examined, or in how health-related quality of life was operationalized. Additionally, because both of these studies focused on older individuals, it is unclear how or if these results may generalize to younger individuals. A recent study among young adults aged 18 to 29 who had engaged in gambling at least once in the preceding year found that impulsivity, a personality trait distinct from but related to inhibitory control deficits [45], was associated with lower self-reported quality of life; however, there was no significant association between performance on an inhibitory control task (stop-signal task) and quality of life in this sample [46]. Thus, the relationship between inhibitory control and health-related quality of life, particularly among adolescents and young adults, remains to be elucidated.

1.3 Generalized additive (mixed) models

1.3.1 Generalized linear models

The generalized linear model (GLM) [47] is a widely-utilized statistical tool for regression and classification problems. A GLM consists of three components, namely, a random component, a systematic component, and a link function. The random component specifies the distribution of the response variable, conditional on the values of the explanatory variables in the model; traditionally, this distribution belongs to the exponential family. The systematic component, or linear predictor, is a linear combination of the explanatory variables. Finally, the link function relates the expected value of the response variable to the linear predictor. Different link functions may be used depending on the distribution of the response variable (e.g., the identity link is traditionally used for normally-distributed data, as is the logit link for binomially-distributed data) [47, 48]. The GLM has the following general form

$$g(\mu) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p \tag{1.1}$$

where $g(\cdot)$ represents the link function; $\mu = E(Y)$ is the conditional expectation of the response variable Y, and x_j represents the j^{th} of p covariates. Estimates for the unknown regression parameters, β_j, and standard errors are obtained through maximum likelihood procedures [48].

A central assumption of the GLM is that of a linear relationship between the response variable (via the link function) and the explanatory variable(s). However, this assumption is often violated. When the relationship between the response variable and an explanatory variable is not linear, a commonly used and relatively straightforward approach is to introduce polynomial terms (e.g., quadratic, cubic) into the model in an attempt to better capture this nonlinearity. However, this approach has several disadvantages. Specifically, the inclusion of polynomial terms may introduce problems with collinearity [49] and result in overfitting [50]. Additionally, this method requires the researcher to make a decision about the shape of the observed relationship, which is rarely if ever truly known. This is also a limitation of other methods such as growth curve modeling that *a priori* specify a functional form for the relationship between the response and explanatory variables (e.g., the Gompertz function [51]). Together, this suggests that a more flexible approach may be beneficial when modeling complex, nonlinear relationships.

1.3.2 Generalized additive models

The generalized additive model (GAM) [52] is a flexible, semiparametric method for identifying and estimating nonlinear effects of explanatory variables on the response variable.

$$g(\mu) = \beta_0 + s_1(x_1) + ... + s_p(x_p) \tag{1.2}$$

Here, $g(\cdot)$ represents the link function; $\mu = E(Y)$ is the conditional expectation of the response variable Y, x_j represents the value of the j^{th} of p covariates, and $s_j(\cdot)$ represents an unknown smooth function of covariate x_j. The smooth functions $s_j(\cdot)$ can be estimated using a number of different methods, including running means, kernel estimates, or splines [52]. However, splines are most commonly used to represent these smooth functions [53, 54]. Using splines, each smooth function $s_j(x_j)$ can be represented by a sum of basis functions

$$s_j(x_j) = \sum_{k}^{K_j} \beta_{jk} b_{jk}(x_j) \tag{1.3}$$

where K_j is the basis dimension (the number of basis functions in the set), $b_{jk}(x_j)$ are specified basis functions, and β_{jk} are parameters to be estimated [53]. Penalized maximum likelihood is used to estimate these coefficients, and in practice this is typically accomplished using penalized iteratively re-weighted least squares. The penalty terms, or smoothing parameters $\boldsymbol{\lambda} = \lambda_1, ..., \lambda_j$ balance smoothness and flexibility [53]. If $\boldsymbol{\lambda}$ is too large, oversmoothing occurs, while, conversely, if $\boldsymbol{\lambda}$ is too small, overfitting occurs. Thus, it is crucial to choose optimal values for $\boldsymbol{\lambda}$, and this can be accomplished using generalized cross validation [53].

There are several broad classes of splines that can be employed in GAMs, including regression splines, natural splines, and smoothing splines. Regression and natural splines generate a curve composed of k segments, which are constructed in between $k+1$ points, or "knots;" the k individual curves are joined together at these knots such that the entire curve is continuous [53]. However, a disadvantage of these spline methods is that they require the researcher to specify the number and spacing of these knots, which can be problematic because model fit often depends on knot placement [53].

Smoothing splines, particularly thin plate splines, are in many ways optimal smoothers. Because thin plate splines place a knot at every data point, they do not require the specification of the number or location of knots as do regression or natural splines. Thin plate splines are often referred to as "full-rank" because the basis dimension is equal to the number of observations [55].

However, as a result, the number of unknown parameters to be estimated is equal to the number of observations, and thus there is a significant associated computational cost [53, 55, 56]. Thin plate regression splines [56] offer an attractive alternative to thin plate splines. Thin plate regression splines approximate thin plate splines by "truncating" the full-rank basis of thin plate splines in order to obtain a lower-rank smoother, retaining the advantages thin plate splines provide (e.g., no knot placement) while improving computational efficiency [53, 56]. Thin plate regression splines are the default basis for smooths in a popular R package for fitting GAMs [57].

1.3.3 Generalized additive mixed models

One important assumption of both GLMs and GAMs is that observations are independent. However, this assumption is violated in the case of longitudinal data, in which repeated measurements are acquired over time from a sample of participants. In this case, additional methods are needed that can account for the correlation between observations from a given participant.

Mixed effects regression models, which incorporate both fixed and random effects, are important tools for longitudinal data analysis as they account for this intra-subject correlation, and have several advantages over other approaches (e.g., generalized estimating equations). First, participants are not required to have the same number of observations, thus allowing participants with incomplete data to be included in analyses. Participants are also not required to have been measured at the same time points, which is particularly useful in longitudinal studies where follow-up times may, and likely do, vary across participants. Finally, mixed effects regression approaches allow for the estimation of individual-level change in addition to the estimation of average or population-level change [58]. Just as generalized linear mixed models (GLMMs) were developed as mixed model extension of GLMs, generalized additive mixed models (GAMM) [59] were developed as a mixed model extension of GAMs. GAMMs have the following general form

$$g(\mu_{it}) = \beta_0 + s_1(x_{1it}) + ... + s_p(x_{pit}) + z'_{it}b_i \tag{1.4}$$

where $\mu_{it} = E(Y_{it})$ is the conditional expectation of the response variable Y for subject i, $i = 1, ..., n$ at time t, $t = 1, ..., T_i$; $g(\cdot)$ is the link function; $s_j(\cdot)$ are the unknown smooth functions; x_{jit} is the j^{th} of p covariates associated with fixed effects; z'_{it} is a vector of q covariates associated with random effects; and b_i is a vector of random effects [60].

Three possible types of random effects can be specified in GAMMs, including random intercepts, slopes, and smooths. Random intercepts characterize subject-specific deviation from the mean or population-average response, while random slopes characterize subject-specific deviation from the population-average covariate effect over time [58]. Although random intercepts and slopes can also be specified in GLMMs, GAMMs are unique in that they additionally allow for random smooths [53, 61]. Random smooths are similar to random slopes, but are more flexible. In GAMMs, random slopes essentially "rotate and stretch" the same curve to best fit a given subject's trajectory, while random smooths fit a separate curve for each trajectory [61]. However, models with random smooths require $n \times k$ basis functions to be fit and thus may be prohibitively resource-intensive [61].

Previous work from our laboratory characterizing the functional development of brain regions supporting inhibitory control in a longitudinal sample of adolescents and young adults employed GLMMs, using a polynomial term for age to capture nonlinearity [22]. Although this study provided us with important insight into normative neurocognitive developmental trajectories during adolescence, there are some limitations associated with this analytical approach. First, as discussed previously, the use of a polynomial term assumes that nonlinear age effects follow some known form such as a quadratic, cubic, or inverse function. The application of flexible GAMMs to model neurocognitive development may provide us with a more nuanced understanding of the shapes of these trajectories during adolescence. GAMMs have previously been used to model longitudinal changes in brain structure [62–64], but to our knowledge have yet to be employed in modeling longitudinal changes in brain function. Additionally, because GAMMs permit the estimation of random subject effects including random intercepts and slopes, as do GLMMs, as well as random smooths, after model-fitting these estimates can be extracted and used as covariates in subsequent analyses. In this way, we may be able to determine how individual differences in neurocognitive developmental trajectories during adolescence relate to other factors, such as subjective measures of health and well-being in young adulthood.

1.4 Regularized regression methods

In a linear model of the form $Y = X\beta + \epsilon$, where Y is a vector of observed response variable values, X is an $n \times p$ design matrix that contains the values of each of p covariates for each of n observations, and ϵ is a vector of error terms, we may wish to both predict the response variable

based on the covariates, as well as determine which covariates are important for this prediction [65]. The unknown parameters $\beta = \beta_1, ..., \beta_p$ are traditionally estimated through least squares, by minimizing the residual sum of squares (RSS).

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{1.5}$$

When the number of predictors p is less than the sample size n, the method of least squares performs well. Although the least squares estimates have high variance, they are unbiased, and thus mean squared error (MSE), which is equal to the sum of the variance of the estimate, the squared bias of the estimate, and some irreducible error, will be relatively low. A low MSE corresponds to more accurate prediction performance. However, when p approaches n, although bias remains low, the variance of the least squares estimates increases, consequently increasing MSE. Finally, in the case of high-dimensional data where $p > n$, least squares has no unique solution, and any of these solutions will be overfit to the data [65, 66]. Regularized regression or "shrinkage" methods are biased, but are also less flexible and thus have less variance. In situations where $p > n$, unlike least squares, these methods can still perform well by trading a small increase in bias for a reduction in variance, consequently reducing MSE [66].

Problems with relatively high-dimensional data (i.e., $p > n$) may often be encountered in neuroimaging research. Small sample sizes are common in fMRI studies due to the high cost associated with acquiring these data [67]. A recent systematic review of task-based fMRI papers published in 2017 reported a median sample size of only $n = 33$, with almost 75% of studies having 50 or fewer subjects [68]. Thus, regularized regression methods may offer a useful alternative for modeling data from neuroimaging studies, where p can conceivably approach or exceed n.

1.4.1 Ridge regression

In ridge regression [69], the coefficients $\beta_1, ...\beta_p$ are estimated by minimizing the following objective, which is similar to that of least squares, but includes a penalty term.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2 \tag{1.6}$$

The penalty term $\lambda \sum_j^p {\beta_j}^2$ is referred to as the ℓ_2 penalty. The parameter λ is a non-negative tuning parameter that can be determined through cross-validation. When $\lambda = 0$, the estimated

coefficients are the same as those produced by least squares. However, as λ is increased away from zero, the magnitudes of the estimated coefficients, $\hat{\beta}_1, ..., \hat{\beta}_j$, decrease or "shrink" in aggregate toward zero [66]. Although the ℓ_2 penalty shrinks coefficient estimates toward zero, it does not force any estimates to be exactly zero [66]. Thus one notable disadvantage of ridge regression is that it does not perform variable selection; all p covariates will be included in the final model in some capacity. This lack of parsimony can complicate model interpretation, particularly when modeling high-dimensional data.

1.4.2 Lasso regression

Lasso regression [70] is another regularized regression method, which, unlike ridge regression, can perform variable selection. In lasso regression, the regression coefficients $\beta_1, ...\beta_p$ are estimated by minimizing the following objective. Like ridge regression, this objective is similar to that of least squares, but includes a penalty term.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1.7}$$

The lasso penalty term, $\lambda \sum_j^p |\beta_j|$, is referred to as the ℓ_1 penalty. Unlike with the ℓ_2 penalty of ridge regression, with the ℓ_1 penalty (assuming a sufficiently large value for λ), some coefficient estimates are forced to be exactly zero [66]. Thus, lasso regression can perform variable selection and yield a "sparse" model. The subset of covariates included in the final model depends on the value of λ; as the value of λ increases, more coefficients will shrink to zero and thus fewer covariates will be included.

Although lasso regression is advantageous for performing variable selection, it has some limitations. Ridge regression typically performs better than does lasso regression when there are fewer covariates than observations (i.e., when $p < n$) [66]. In the case of high-dimensional data where $p > n$, lasso regression can only select at most n covariates before it saturates [71]. Additionally, if a group of covariates is highly correlated, lasso regression tends to arbitrarily choose one of them and exclude the others [71].

1.4.3 Elastic net regression

Elastic net regression [71] was recently developed in an attempt to overcome the individual limitations of both ridge and lasso regression approaches. The elastic net method utilizes a linear combination of the ℓ_2 penalty of ridge regression and the ℓ_1 penalty of lasso regression. In elastic net regression, coefficients are estimated by minimizing the following objective.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \; + \; \lambda_1 \sum_{j=1}^{p}|\beta_j| \; + \; \lambda_2 \sum_{j=1}^{p}{\beta_j}^2 \tag{1.8}$$

The solution to the above is referred to as the "naive" elastic net solution. The naive elastic net follows a sequential procedure, in which the ridge regression coefficients are determined for a fixed value of λ_2, and then the lasso shrinkage is subsequently performed. However, this procedure results in excessive shrinkage, which increases bias and has little impact on variance, resulting in poorer prediction performance. To mitigate this problem, the naive elastic net estimates are rescaled by a factor of $(1 + \lambda_2)$.

$$\hat{\beta}_{\text{elastic net}} = (1 + \lambda_2)\ \hat{\beta}_{\text{naive elastic net}} \tag{1.9}$$

This transformation preserves the variable selection property of the lasso step while undoing excessive shrinkage [71]. Additionally, elastic net is strictly convex when $\lambda_2 > 0$, which allows for groups of highly correlated predictors to have similar or equal regression coefficients [71]. Thus groups of highly related predictors will tend to be all included in or all excluded from the model, depending on the magnitudes of the coefficients. Elastic net has been shown to perform better in terms of prediction accuracy and sensitivity compared to ridge and lasso methods [72, 73].

1.5 Objectives

The present study aimed to establish the relationship between individual differences in the development of inhibitory control and its neural substrates in adolescence and subsequent health-related quality of life in adulthood. We used data from an accelerated longitudinal cohort of healthy individuals, initially aged 8 to 30 years, who completed up to 13 study visits at approximately yearly intervals. At each study visit, participants performed the antisaccade task, a well-established oculomotor task probing inhibitory control, while fMRI data were acquired. At the conclusion of the

study, participants completed the World Health Organization Quality of Life (WHO-QOL) questionnaire, a self-report measure assessing subjective health and well-being in physical, psychological, social, and environmental domains.

There were two major objectives of the present study: (1) To characterize the normative adolescent development of inhibitory control (as measured by performance on the antisaccade task), as well as activation in brain regions supporting inhibitory control, in our longitudinal sample using GAMMs; and (2) To determine how individual differences in these adolescent developmental trajectories (i.e., random intercepts and slopes) may predict subsequent health-related quality of life in early adulthood using elastic net regression.

2.0 Methods

2.1 Participants

One-hundred and sixty-seven participants (93 female) were studied in an accelerated longitudinal cohort design, in which participants spanning a broad cross-section of age were recruited and followed prospectively. Participants were initially aged 8 to 30 years and were followed for up to 13 time points (mean = 3.65, SD = 2.94) at approximately yearly intervals. At each time point, participants completed an fMRI scan and a separate behavioral session consisting of neuropsychological testing, computerized cognitive tasks, and surveys. Participants were healthy individuals without history of head injury with a loss of consciousness exceeding one hour; vision or eye-movement problems not corrected in childhood; Tourette syndrome or tic disorders; seizures; meningitis; encephalitis; diabetes; genetic disorder; or a personal or family history of psychiatric disorders including autism spectrum disorders, attention deficit hyperactivity disorder, bipolar disorder, major depression, schizophrenia, or any major neurological disorder such as Huntington's chorea. Participants also had no contraindications for scanning such as non-removable metal, weight greater than 250 pounds, or current pregnancy. IQ was measured using the Weschler Abbreviated Scale of Intelligence [74], and all participants had a full-scale IQ greater than 80. The study was approved by the Institutional Review Board at the University of Pittsburgh. All participants, and parents of minor participants, provided written informed consent. Participants were compensated for their time.

Over the course of the study, 41 participants were dropped due to diagnosis of a major neurological or psychiatric disorder in a first-degree relative ($n = 15$), claustrophobia or unwillingness to scan ($n = 11$), poor performance ($n = 8$), brain abnormality ($n = 3$), diagnosis of a psychiatric disorder or use of psychiatric medication ($n = 3$), or seizure episodes ($n = 1$), resulting in an eligible sample of 126 participants. At the conclusion of the study, participants were contacted to complete a brief follow-up study consisting of the World Health Organization Quality of Life (WHO-QOL) questionnaire. Of the participants contacted for the follow-up study, 71 completed the questionnaire. Among these 71 participants, 21 were excluded from the present analyses because they had usable data for fewer than three time points, the minimum number required to allow for the identification of nonlinear age effects. This resulted in a final sample size of 50 participants

(30 female) with a total of 316 scans (Figure 1). On average, each participant completed 6.32 scans (SD = 2.71, min = 3, max = 13), with an average between-scan interval of 1.65 years (SD = 1.58, min = 0.41, max = 10.16).

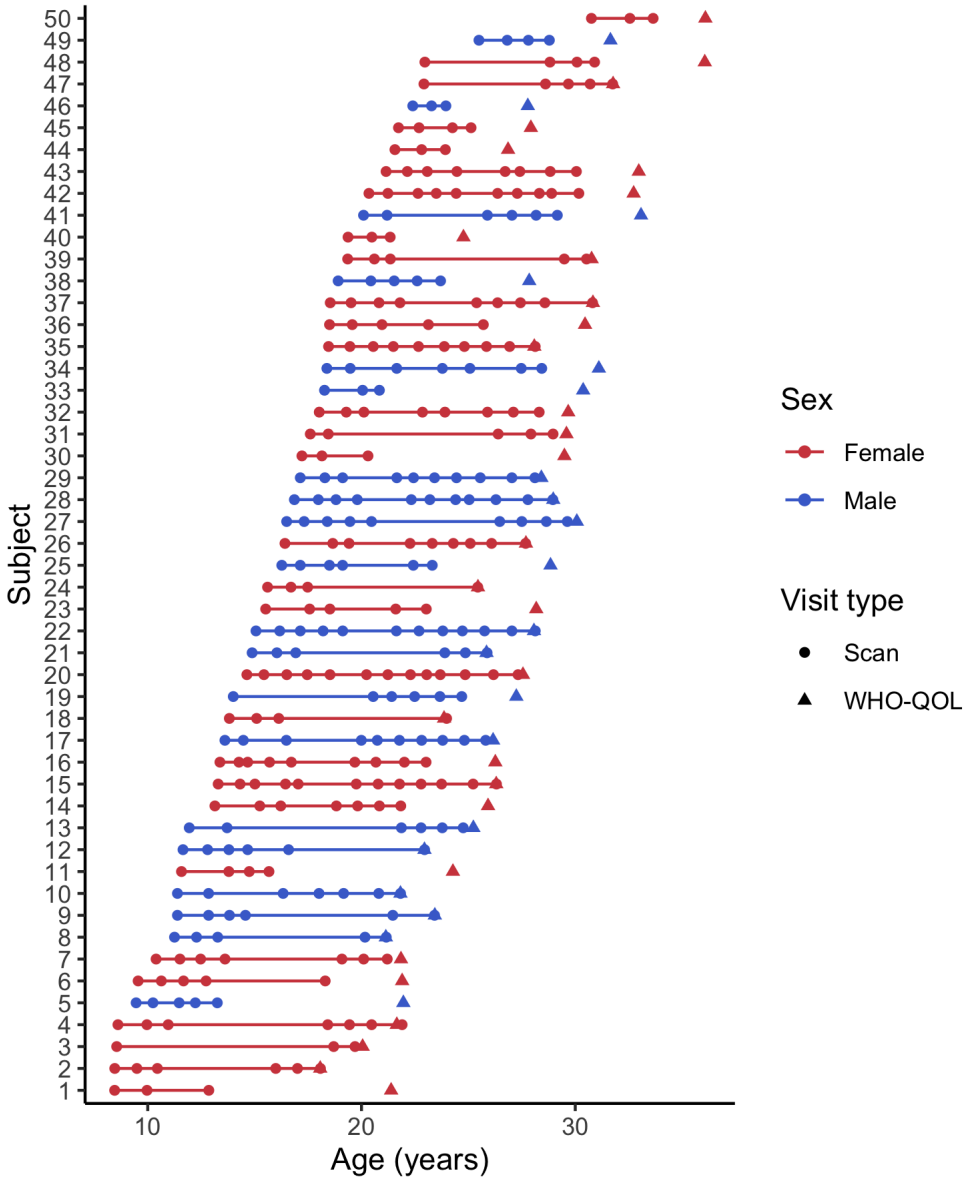

Figure 1: Age distribution of the final sample. Lines interconnect scans for a single participant, where each circle denotes a scan acquisition. For each participant, the left-most circle corresponds to age at baseline scan. Triangles correspond to age at WHO-QOL administration for each participant

2.2 Experimental procedure and data acquisition

2.2.1 Antisaccade task

At each visit, participants completed a total of four task runs in the scanner. Each task run consisted of a block of the antisaccade task and a block of a prosaccade task, each of which was preceded and followed by a block of fixation (i.e., a central cross to which the participant was instructed to look). The order in which the antisaccade and prosaccade task blocks were presented within each run was counterbalanced across runs and participants. In each task run, participants completed 12 trials of the antisaccade task and 12 trials of the prosaccade task, for a total of 48 trials of each type in the experiment. On antisaccade trials, participants first fixated on a red central fixation cross for 3 s. After 3 s, the central fixation disappeared and the saccade target stimulus, a yellow circle, immediately appeared in one of six possible locations in the periphery (presented on the horizontal meridian at $\pm$ 3, 6, or 9° visual angle) for 1.5 s. Stimulus location order was randomized within each task run. In order to perform the antisaccade task, participants were instructed to inhibit the prepotent prosaccade to the stimulus, and instead make an antisaccade to the opposite or mirror-image location. Inter-trial intervals consisted of a white central fixation cross and varied in duration between 3 and 9 s in order to permit estimation of trial-related activation. The procedure for the prosaccade task was identical, with the exception that each trial began with a green central fixation cross, and participants were instructed to make the prepotent prosaccade to the stimulus location when the stimulus appeared. Only the antisaccade task is considered in the present analyses. A PC running E-Prime software was used to control stimulus display. Stimuli were projected onto a translucent screen affixed behind the scanner bore, visible to the participant via a mirror attached to the head coil.

2.2.2 Eye-tracking data acquisition and preprocessing

In order to assess antisaccade task performance, eye movement data were obtained during scanning using a long-range optics eye tracking system (model R-LRO6, Applied Science Laboratories, Bedford, MA) with a sampling rate of 60 Hz. Lights in the scanner room were dimmed to reduce glare and maximize pupil size. The eye tracker field of view was manually centered on the participant's pupil before beginning the scan. Immediately before the task, the eye tracker was calibrated using an evenly-distributed, nine-point stimulus grid. Calibration verification was per-

formed by instructing the participant to re-scan the nine-point grid, and adjustments were made by the experimenter when necessary.

Eye movement data were scored offline with an in-house, automated scoring program using tools from ILAB [75]. Saccades were identified using a velocity algorithm with a 20°/s criterion. Saccades following the presentation of each stimulus were subsequently scored as correct (the participant successfully inhibited the prosaccade and made the antisaccade), error-corrected (the participant made the antisaccade after initially failing to inhibit the prosaccade), incorrect (the participant failed to inhibit the prosaccade), or dropped (unable to be scored, e.g., due to signal dropout or excessive blinks). Primary measures of interest included the proportions of correct and error-corrected antisaccade trials among those trials that were not dropped, as well as the average saccade latency, measured as the time (in ms) from the stimulus onset to the onset of the saccade, for each of these trial types.

2.2.3 Neuroimaging data acquisition and preprocessing

Neuroimaging data were acquired on a 3 Tesla (3T) Siemens Allegra scanner with a standard 8-channel, radio-frequency (RF) head coil at the Neuroscience Imaging Center at the University of Pittsburgh. A saggital magnetization-prepared rapid gradient echo (MPRAGE) T1-weighted pulse sequence (repetition time (TR) = 1630 ms, echo time (TE) = 2.48 ms, flip angle = 8°, voxel size = 0.8 mm isotropic) was used to obtain structural images with which to coregister functional images. Functional images were obtained using an echo-planar imaging (EPI) sequence sensitive to BOLD contrast (TR = 1500 ms, TE = 25 ms, flip angle = 70°, voxel size = 3.1 × 3.1 × 4 mm, 29 contiguous axial slices aligned parallel to the anterior-posterior commissure (AC-PC) plane acquired during each TR). Each functional run was 6 min, 12 s in duration.

Data preprocessing followed our laboratory's standard pipeline, incorporating tools from FMRIB Software Library (FSL) and Analysis of Functional Neuroimages (AFNI) [76]. Functional images were corrected for slice-timing, and motion using a rigid-body rotation and translation algorithm. Functional and structural images were coregistered and aligned to the Montreal Neurological Institute (MNI) 2.3 mm template using a series of affine and nonlinear transformations. Functional images were spatially smoothed using a weighted 5 mm full-width at half-maximum Gaussian kernel, and a 0.025 Hz high-pass temporal filter was applied to remove low-frequency signal drifts due to noise. Finally, the voxel time series was normalized to permit calculation of

percent signal change. Preprocessed fMRI data were analyzed using voxelwise GLMs performed in AFNI. These analyses were performed to estimate changes in the BOLD signal associated with correct and error-corrected antisaccade trials relative to baseline fixation trials in each voxel. Nuisance regressors included baseline signal drifts, mean white matter and cerebrospinal fluid time courses, and six standard motion parameters (translations along the x, y, and z axes, and roll, pitch, and yaw rotations). Subject motion was also addressed by censoring TR values with significant motion.

We examined activation in the same brain regions of interest (ROIs) reported in Ordaz and colleagues [22]. These ROIs included canonical regions involved in inhibitory control, but not specific to the antisaccade task. In particular, this encompassed executive regions including the bilateral dorsolateral and ventrolateral prefrontal cortex (dlPFC and vlPFC, respectively); motor response regions including the supplementary eye fields (SEF), pre-supplementary motor area (pre-SMA), and bilateral frontal eye fields (FEF), posterior parietal cortex (pPC), putamen; and performance and error monitoring regions including the dorsal anterior cingulate cortex (dACC). ROIs were created based on coordinates obtained from Neurosynth [77], a platform for reverse-inference meta-analyses of fMRI studies. ROIs were defined as voxels within a 7-12 mm radius of the respective mean peak coordinates obtained from Neurosynth. The previously estimated beta coefficients reflecting the response magnitude for each voxel within each ROI were averaged to produce a mean percent signal change measure for each ROI.

2.2.4 WHO-QOL questionnaire

Participants completed the WHO-QOL questionnaire in-person at their final study visit, or, for those who were no longer regularly participating (e.g., due to relocation), the questionnaire was completed via an online survey platform. The WHO-QOL questionnaire is a self-report measure assessing subjective health and well-being in physical, psychological, social, and environmental domains. The questionnaire contains 26, 5-point Likert scale items. The first item is an overall rating of the participant's quality of life, and the second item is an overall rating of the participant's satisfaction with their health. The subsequent items contribute to the calculation of four domain-specific quality of life scores. Domain-specific scores were calculated from the raw questionnaire data according to scoring guidelines in the WHO-QOL manual. This included replacing missing values for items corresponding to a given domain with the mean of the participant's responses to the other items in the same domain, and summing response values across individual items

corresponding to each domain in order to obtain four domain-specific quality of life scores (namely, physical, psychological, social, and environmental) for each participant. Domain-specific scores were z-scored such that all domains were on comparable scales. This was necessary because the number of items and the range of possible scores varied across each of the four domains, and thus domain scores were not directly comparable. These four standardized domain-specific scores were then averaged for each participant to create an overall or composite measure of health-related quality of life for each individual.

2.3 Statistical analyses

2.3.1 Generalized additive mixed models

The first objective of this thesis was to characterize the normative adolescent development of inhibitory control, and the brain regions that support this cognitive process, in our longitudinal sample. Outcome variables of interest were modeled separately and included each of the following behavioral and brain measures: proportion correct antisaccade trials, proportion error-corrected antisaccade trials, average saccade latency on correct trials, average saccade latency on error-corrected trials, activation (percent BOLD signal change) during correct trials in dlPFC, vlPFC, SEF, pre-SMA, FEF, pPC, putamen, dACC, and activation during error-corrected trials in dACC. The primary covariate of interest in each model was age, which we expected, based on previous findings, may exhibit a nonlinear relationship with each outcome. Age was centered to permit meaningful interpretation of model intercept terms. Given that there are notable between-sex differences in brain structure and function [78], and that maternal education is a strong predictor of brain and cognitive development [79, 80], we assessed whether the potential confounding variables of sex (female, male) and maternal education level at the time of the subject's baseline scan (categorized as completed high school, college, or graduate/professional school) should be controlled for in each model.

The generalized additive mixed model (GAMM) [59] is a flexible, semiparametric method for identifying and estimating nonlinear effects of covariates on the outcome variable when observations

are not independent, such as with longitudinal data. The key feature of GAMMs is that the mean of the outcome variable depends on the covariates through a sum of smooth terms [53, 56].

$$g(\mu_{it}) = \beta_0 + s_1(x_{1it}) + ... + s_p(x_{pit}) + z'_{it}b_i \tag{2.1}$$

where $\mu_{it} = E(Y_{it})$ is the conditional expectation of the response variable Y for subject i, $i = 1, ..., n$ at time t, $t = 1, ..., T_i$; $g(\cdot)$ is the link function; $s_j(\cdot)$ are unknown smooth functions to be estimated; x_{jit} is the j^{th} of p covariates associated with fixed effects; z'_{it} is a vector of q covariates associated with random effects; and b_i is a vector of random effects [60]. Splines are most commonly used to represent these smooth functions [53, 54]. Using splines, each smooth function $s_j(x_j)$ can be represented by a linear combination of a set of simpler functions that have a known, closed form. These functions are referred to as basis functions.

$$s_j(x_j) = \sum_{k}^{K_j} \beta_{jk} b_{jk}(x_j) \tag{2.2}$$

where K_j is the basis dimension (the number of basis functions in the set), $b_{jk}(x_j)$ are specified basis functions, and β_{jk} are parameters to be estimated [53]. Penalized maximum likelihood is used to estimate these coefficients, and in practice this is typically accomplished using penalized iteratively re-weighted least squares. The penalty terms, or smoothing parameters $\boldsymbol{\lambda} = \lambda_1, ..., \lambda_j$ balance smoothness and flexibility [53]. If $\boldsymbol{\lambda}$ is too large, oversmoothing occurs, while, conversely, if $\boldsymbol{\lambda}$ is too small, overfitting occurs. Thus, it is crucial to choose optimal values for $\boldsymbol{\lambda}$, and this can be accomplished using generalized cross validation [53].

There are several broad classes of splines that can be employed in GAMMs, including regression splines, natural splines, and smoothing splines. Regression and natural splines generate a curve composed of k segments, which are constructed in between $k+1$ points, or "knots;" the k individual curves are joined together at these knots such that the entire curve is continuous [53]. However, a disadvantage of these spline methods is that they require the researcher to specify the number and spacing of these knots, which can be problematic because model fit often depends on knot placement [53].

Smoothing splines, particularly thin plate splines, are optimal smoothers. Because thin plate splines place a knot at every data point, they do not require the specification of the number or location of knots as do regression or natural splines. Thin plate splines are often referred to as "full-rank" because the basis dimension is equal to the number of observations [55]. However, as

a result, or a sample of n data points, thin plate splines require n parameters plus an additional smoothing parameter to be estimated, and thus there is a significant associated computational cost [53, 56]. Thin plate regression splines [56] offer an attractive alternative to thin plate splines. Thin plate regression splines approximate thin plate splines by "truncating" the full-rank basis of thin plate splines in order to obtain a lower-rank smoother, retaining the advantages thin plate splines provide (e.g., no knot placement) while improving computational efficiency [53, 56].

In GAMMs, three types of random effects are permitted to be specified. These include random intercepts, which characterize subject-specific deviation from the mean or population-average response; random slopes, which characterize subject-specific deviation from the population-average covariate effect over time; and random or factor smooths, which characterize subject-specific smooth functions of the covariate; these smooths share the same smoothing parameter λ but may have different shapes [53, 61]. As the random smooths may inherently capture differences in intercepts or slopes, random smooths for a given covariate are generally not used in conjunction with random intercepts and slopes for that same covariate [61, 81].

In the present analyses, the *mgcv* package for R (version 1.8-31 [57]) was used to fit a series of GAMMs for the outcomes of interest, each with a smooth function of age as a covariate, using a thin plate regression spline basis to estimate this smooth function. Random effects in each GAMM included subject-specific intercepts and slopes for age. Because the antisaccade accuracy measures are proportions, the quasi-binomial family and logit link function was used for these outcomes. For all other outcomes, which are continuous, a Gaussian family and identity link function was used. When evaluating the significance of each smooth term for age, the Benjamini-Hochberg procedure [82] was used to account for multiple comparisons by controlling the false discovery rate. However, because the aim of this set of analyses is to characterize development, age was retained in each model even if it did not have a statistically significant effect on the respective outcome measure. Each model was also refit including additional fixed effects for the potential confounders of sex and maternal education level. The significance of these covariates, determined by t tests (sex) or partial F tests (maternal education level), were used as criteria to determine if these confounders should be retained in each of the final models.

For those models for which there was a statistically significant effect of age, the *gratia* package for R (version 0.3.0 [83]) was used to conduct exploratory post-hoc analyses to identify significant periods of developmental change. Specifically, the derivatives of each estimated smooth function of age were approximated using the method of finite differences, and a simultaneous 95% confidence

band for the derivatives were calculated. The age range(s) for which each 95% confidence band did not contain zero was considered to represent a period of significant developmental change. The random intercepts and slopes from each of these final models were extracted and saved for use in subsequent analyses.

2.3.2 Bootstrap-enhanced elastic net regression

The second objective of this thesis was to determine how individual differences in the adolescent development of inhibitory control and brain regions supporting this cognitive process predict subsequent health-related quality of life in early adulthood. The outcome measure for this analysis was the standardized composite quality of life score obtained by averaging the four standardized domain-specific scores from the WHO-QOL questionnaire. Covariates included random slopes and intercepts extracted from those GAMM models for which there was a significant effect of age. Random intercepts are considered to represent person-specific deviations from the mean level of the respective measure of inhibitory control at the mean age of the sample, while random slopes are considered to represent person-specific deviations in the rate of development of that measure. Because potential confounders were controlled for at the GAMM step, no confounding variables were controlled for in this portion of the analysis.

Because the number of predictors ($p_{\max} = 36$) could potentially have approached the size of our sample ($n = 50$), a regularized regression method was appropriate. Elastic net regression is a regularized regression method that utilizes a linear combination of the ℓ_2 penalty of ridge regression and the ℓ_1 penalty of lasso regression. In naive elastic net regression, coefficients are estimated by minimizing the following objective.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} {\beta_j}^2 \tag{2.3}$$

where y_i represents the response for the i^{th} of n subjects, x_{ij} represents the value of the j^{th} of p covariates for subject i, λ_1 is the tuning parameter corresponding to the ℓ_1 lasso penalty, and λ_2 is the tuning parameter corresponding to the ℓ_2 ridge penalty. The ℓ_1 portion of the elastic net penalty generates a sparse model (i.e., allows for variable selection), while the ℓ_2 portion encourages a grouping effect by which highly correlated predictors tend to be included or excluded from the model in aggregate [71].

The above objective can be reparameterized as the following.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \;+\; \lambda\left(\frac{1-\alpha}{2}\;\sum_{j=1}^{p}{\beta_j}^2 \;+\; \alpha\;\sum_{j=1}^{p}|\beta_j|\right) \tag{2.4}$$

where $\alpha \in [0,1]$ is a "mixing" parameter [71]. When $\alpha = 0$, the solution to the above is equivalent to that of ridge regression, and when $\alpha = 1$, the solution to the above is equivalent to the that of lasso regression. Thus the α parameter controls the relative contributions of the ridge and lasso penalty terms. The tuning parameter λ controls the overall strength of the elastic net penalty [71]. Optimal values for these hyperparameters can be determined using k-fold cross validation. In k-fold cross validation, the dataset is randomly split into k partitions or "folds," where k is typically equal to 5 or 10. On each of the k iterations, $k-1$ folds are used in combination to train the model, and the remaining fold is reserved as a test fold to validate the model. A cross-validated accuracy measure can be obtained by averaging the mean squared errors (MSE) computed from each of the test folds [66].

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i \tag{2.5}$$

Using grid search, the k-fold cross validation procedure is repeated for each possible combination of α and λ values. The pair of α and λ values that result in the minimum cross-validated error are considered the optimal hyperparameter values.

One disadvantage associated with regularized regression approaches, including elastic net regression, is that confidence intervals and p-values are generally not available for the coefficients estimated by these methods, and thus inference is not possible [84]. However, a bootstrap-enhanced elastic net method was developed by Bunea and colleagues [85], and recently expanded by Abram and colleagues [86], in order to overcome this issue. In the bootstrap-enhanced elastic net method, B bootstrap samples of size n are randomly sampled from the data with replacement. The elastic net model is then refit separately to each of the B bootstrap samples to produce B different estimates of each of the coefficients. The distributions of these coefficient estimates (i.e., bootstrap distributions) can then be used to assess the importance of the associated predictors, by one of two approaches.

In the quantile approach [86], the quantiles of the bootstrap distribution of β_j are used to determine the significance of the j^{th} predictor. Assuming a significance level α^*, the $100(1-\alpha^*)\%$ bootstrap confidence interval for β_j is given by $\left[Q_{j,\alpha^*/2},\; Q_{j,1-\alpha^*/2}\right]$, where $Q_{j,\alpha}$ is the quantile value

such that $\alpha = \frac{1}{B}\sum_{b=1}^{B} I(\hat{\beta}_{jb} \leq Q_{j,\alpha})$, where $I(\hat{\beta}_{jb} \leq Q_{j,\alpha})$ is an indicator function that equals 1 if $\hat{\beta}_{jb} \leq Q_{j,\alpha}$ and equals 0 otherwise, and $\hat{\beta}_{jb}$ is the estimated coefficient for the j^{th} predictor on the b^{th} bootstrap iteration [86]. If the $100(1-\alpha^*)\%$ bootstrap confidence interval for a given coefficient does not contain zero, the associated variable is considered to be a statistically significant predictor of the outcome.

Alternatively, the variable inclusion probability (VIP) approach [85] can be utilized. The VIP quantifies the importance of a given predictor as the proportion of the B iterations of the bootstrap procedure on which the predictor received a nonzero coefficient estimate. Assuming the significance level α^*, a VIP greater than $(1-\alpha^*)\%$ indicates that a variable is a statistically significant predictor. The VIP may also be interpreted in a Bayesian context. Specifically, assuming a Laplace prior on β_j, the VIP reflects the posterior probability of the j^{th} predictor being included in the model [85].

$$\text{VIP}_j = P(\beta_j \neq 0 \mid \text{Data}),\ j = 1, ..., p \tag{2.6}$$

The bootstrap-enhanced elastic net procedure was utilized in the present analyses because it allows for automatic variable selection and also provides methods for determining which variables are important or statistically significant predictors of the outcome. The *ensr* (version 0.1.0 [87]) and *glmnet* (version 3.0-2 [88]) packages for R were used to tune hyperparameters and fit the optimal elastic net regression model, respectively. Grid search was used in conjunction with ten-fold cross validation to perform simultaneous hyperparameter tuning. The elastic net model was fit using the pair of hyperparameter values which minimized the cross-validated mean squared error (CV-MSE). The model was then refit on $B = 5000$ bootstrap samples of size $n = 50$ to obtain bootstrap distributions for each of the estimated coefficients. The 2.5^{th} and 97.5^{th} percentiles of each distribution were calculated and used to construct 95% bootstrap confidence intervals for each coefficient. Those predictors for which the 95% bootstrap confidence interval did not contain zero were considered statistically significant. Additionally, variable inclusion probabilities (VIP) were calculated for each predictor as the proportion of the 5000 bootstrap iterations on which the predictor received a nonzero coefficient estimate. A VIP greater than or equal to 95% was also considered as a criterion for statistical significance. To visualize and better understand potential associations between predictors and health-related quality of life, a post-hoc analysis included performing a median split to categorize participants into high and low quality of life groups. Individual GAMMs were then refit including a smooth age by quality of life group interaction term, and estimated developmental trajectories for each quality of life group were plotted.

3.0 Results

3.1 Characterizing adolescent development of inhibitory control

The first objective of the present analysis was to characterize the normative adolescent development of inhibitory control in our longitudinal sample. A separate GAMM was fit with a smooth function of age as a covariate, and subject-specific intercepts and slopes for age included as random effects, for each outcome measure of interest including proportion correct antisaccade trials, proportion error-corrected antisaccade trials, average saccade latency on correct trials, average saccade latency on error-corrected trials, activation (percent BOLD signal change relative to baseline) during correct trials in dlPFC, vlPFC, SEF, pre-SMA, FEF, pPC, putamen, and dACC, and activation during error-corrected trials in dACC. Each GAMM was also refit separately including additional fixed effects for the potential confounders of sex (Table 1) and maternal education level (Table 2) to determine if either or both of these covariates should be used as adjustments in the respective final models. For each final model, the effective degrees of freedom (edf), F statistic, p-value, and false discovery rate adjusted p-value (q-value) for the smooth term for age are reported in Table 3. All statistically significant findings described below survived correction for multiple comparisons (i.e., q-value < 0.05) unless otherwise specified.

3.1.1 Behavioral measures

There were statistically significant effects of age for the proportion of correct antisaccade trials ($p < 0.0001$), average latency on correct antisaccade trials ($p < 0.0001$), and the proportion of error-corrected antisaccade trials ($p < 0.0001$; Table 3). There was not a significant effect of age for average latency on error-corrected antisaccade trials ($p = 0.0550$; Table 3), although age was retained in the model because age is the primary covariate of interest. In the models with age as the covariate of interest, there were no statistically significant effects of sex for the proportion of correct antisaccade trials, the proportion of error-corrected antisaccade trials, average latency on correct antisaccade trials, or average latency on error-corrected antisaccade trials ($p \geq 0.1971$; Table 1). Additionally, in the models with age as the covariate of interest, there were no statistically significant global effects of maternal education level for any of the behavioral measures ($p \geq 0.1080$;

Table 2). Therefore, the final models for each of the behavioral measures included only the smooth term for age (Figure 2). For those outcome measures for which there were significant effects of age, post-hoc analyses indicated that the proportion of correct antisaccade trials increased from age 8.45 to 21.36 years, while average latency on correct antisaccade trials and the proportion of error-corrected antisaccade trials decreased from 8.45 to 23.01 years, and from 11.48 to 20.98 years, respectively.

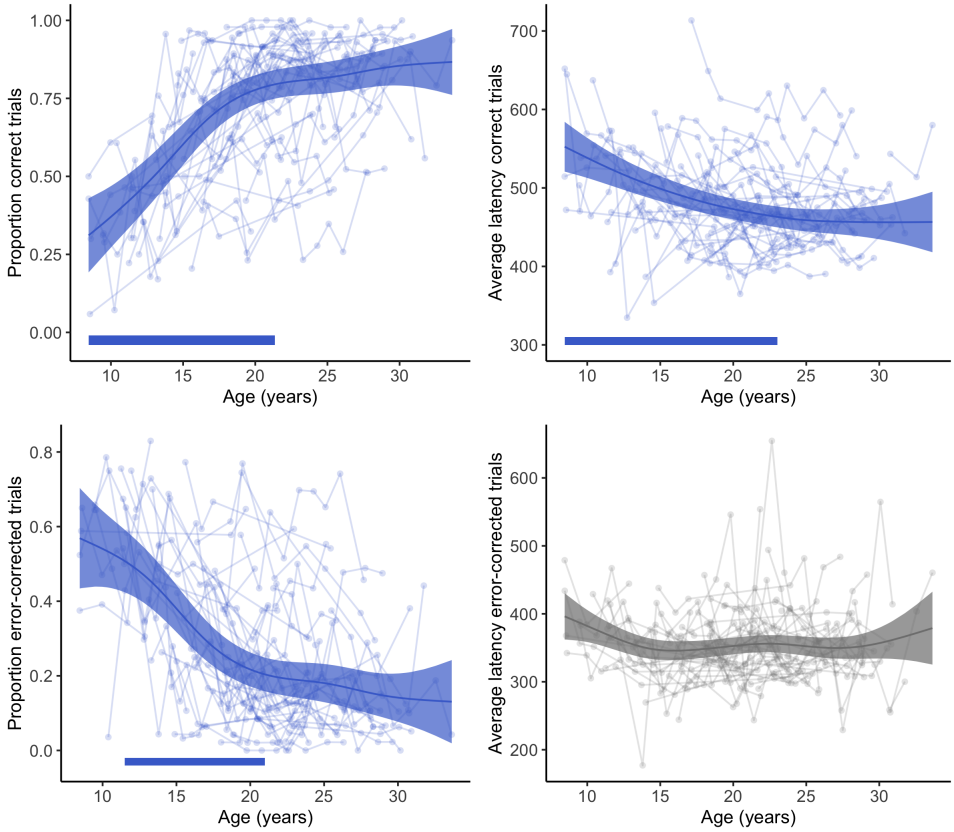

Figure 2: Effects of age for each behavioral measure. A spaghetti plot of individual subject trajectories for each behavioral measure is overlaid with a plot of fitted values and point-wise 95% confidence intervals for the smooth function of age estimated in each respective GAMM. Plots displayed in blue indicate that the age effect was statistically significant, with the bar above the x-axis indicating the period of significant developmental change. Plots displayed in grey indicate that the age effect was not statistically significant. There were significant effects of age for the proportions of correct (top left) and error-corrected (bottom left) antisaccade trials as well as average saccade latency on correct antisaccade trials (top right), continuing into young adulthood

3.1.2 Brain function measures

3.1.2.1 Motor response regions Among the motor response ROIs, there were statistically significant effects of age for activation during correct antisaccade trials in L FEF ($p = 0.0142$), L pPC ($p = 0.0060$), and R pPC ($p = 0.0055$); there were no statistically significant age effects in SEF ($p = 0.9644$), pre-SMA ($p = 0.3244$), R FEF ($p = 0.0785$), L putamen ($p = 0.5025$), or R putamen ($p = 0.7368$; Table 3). In the models with age as the covariate of interest, there were no statistically significant effects of sex ($p \geq 0.0705$; Table 1) or maternal education level ($p \geq 0.5705$; Table 2) on activation in any of the motor response ROIs. Therefore, the final models for activation in each of the motor response ROIs includes only the smooth term for age (Figure 3). For those ROIs for which there were significant age effects, post-hoc analyses indicated that activation decreased from age 10.98 to 16.93 years in L FEF, from 11.11 to 17.31 years in L pPC, and from 9.71 to 16.67 years in R pPC.

3.1.2.2 Executive control regions Among the executive control ROIs, there was a statistically significant effect of age only for activation during correct antisaccade trials in R dlPFC ($p = 0.0064$; Table 3). There were no significant effects of age in L dlPFC ($p = 0.9828$), L vlPFC ($p = 0.7332$), or R vlPFC ($p = 0.0581$; Table 3). Additionally, in the models with age as the covariate of interest, there were no statistically significant effects of either sex ($p \geq 0.1697$; Table 1) or maternal education level ($p \geq 0.3502$; Table 2) in any of the executive control ROIs. Therefore, the final models for activation in each of the executive control ROIs includes only the smooth term for age (Figure 4). For the ROI for which there was a significant age effect, a post-hoc analysis indicated that there were developmental decreases in R dlPFC activation from age 10.60 to 16.67 years.

3.1.2.3 Performance monitoring region In the performance monitoring ROI, dACC, there was not a significant effect of age for activation during correct antisaccade trials ($p = 0.3770$; Table 3). Additionally, in the model with age as the covariate of interest, there were no statistically significant effects of sex ($p = 0.3185$; Table 1) or maternal education level ($p = 0.0847$; Table 2). Therefore, the final model for dACC activation during correct antisaccade trials includes only the smooth term for age (Figure 5).

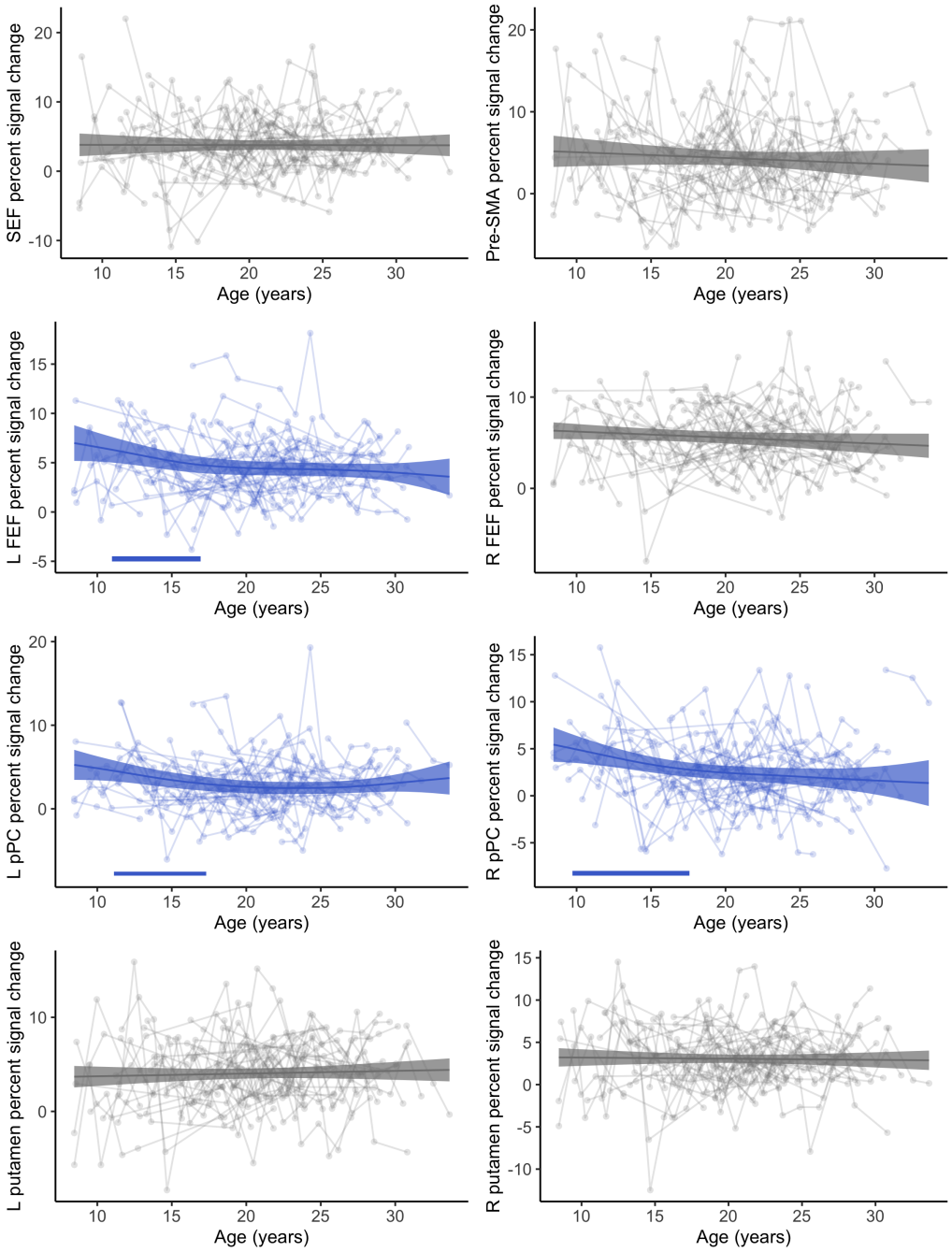

Figure 3: Effects of age for activation during correct antisaccade trials in each motor response region. A spaghetti plot of of individual subject trajectories for activation (percent signal change) in each motor response ROI is overlaid with a plot of fitted values and point-wise 95% confidence intervals for the smooth function of age estimated with the respective final GAMM. There were significant effects of age for activation during correct antisaccade trials in L FEF, L pPC, and R pPC (displayed in blue), with age-related change continuing into mid-adolescence

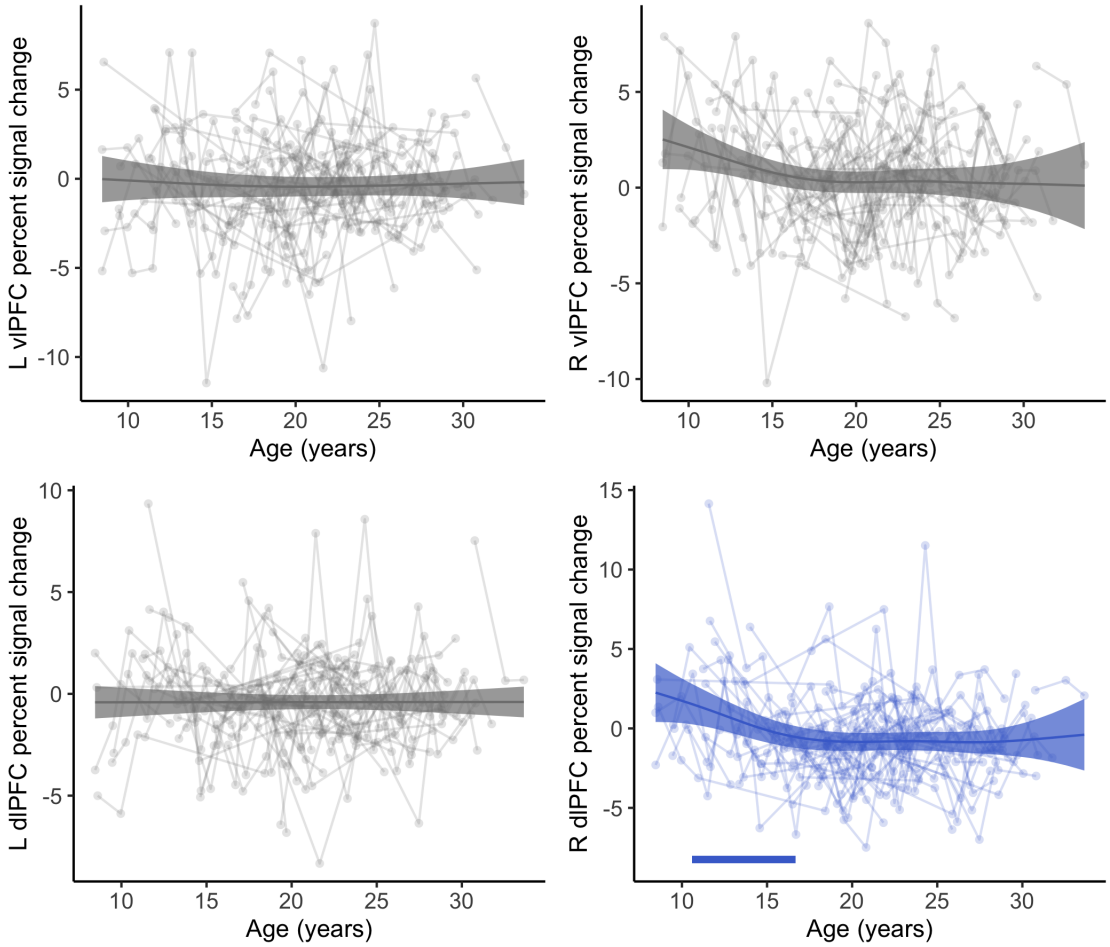

Figure 4: Effects of age for activation during correct antisaccade trials in each executive control region. A spaghetti plot of individual subject trajectories for activation (percent signal change) in each executive control ROI is overlaid with a plot of fitted values and point-wise 95% confidence intervals for the smooth function of age estimated with the respective final GAMM. Plots displayed in blue indicate that the age effect was statistically significant, with the bar above the x-axis indicating the period of significant developmental change. Plots displayed in grey indicate that the age effect was not statistically significant. There was a significant effect of age for activation during correct antisaccade trials only in R dlPFC (bottom right), with age-related change continuing into mid-adolescence

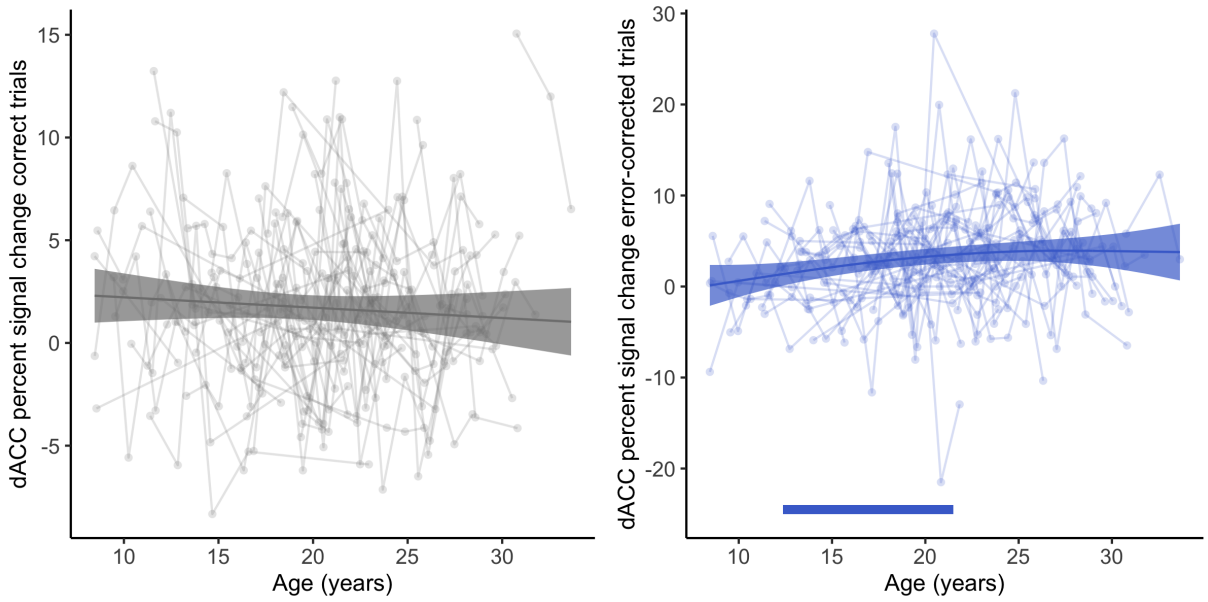

Figure 5: Effects of age for activation in the performance monitoring region, dACC. A spaghetti plot of individual subject trajectories for activation (percent signal change) during correct (left) and error-corrected (right) antisaccade trials in dACC is overlaid with a plot of fitted values and point-wise 95% confidence intervals for the smooth function of age estimated with the respective final GAMM. Adjusting for sex, there was a significant effect of age for dACC activation during error-corrected antisaccade trials, with age-related change continuing into young adulthood

There was, however, a significant effect of age in dACC for activation during error-corrected antisaccade trials ($p = 0.0184$; Table 3). In the model with age as the covariate of interest, there was a significant effect of sex ($p = 0.0161$; Table 1), with average dACC activation during error-corrected antisaccade trials about 1.67% higher in males compared to females. There was not a significant effect of maternal education level ($p = 0.3654$; Table 2). Therefore, the final model for dACC activation during error-corrected antisaccade trials includes both the smooth term for age and sex as covariates (Figure 5); including sex in the model did not substantially change the significance of the smooth term for age ($p = 0.0120$). A post-hoc analysis indicated that, adjusting for sex, there were developmental increases in activation in dACC during error-corrected antisaccade trials from age 12.37 to 21.49 years.

Table 1: Estimated coefficients and significance of sex for each outcome measure, controlling for age

Outcome measure	$\hat{\beta}$	t	p-value
Proportion correct trials	0.0873	0.3424	0.7322
Proportion error-corrected trials	-0.0698	-0.2851	0.7756
Average latency correct trials	-3.9599	-0.3069	0.7590
Average latency error-corrected trials	-12.8374	-1.2925	0.1971
SEF activation correct trials	-1.2471	-1.8144	0.0705
Pre-SMA activation correct trials	0.0283	0.0304	0.9757
L FEF activation correct trials	0.1769	0.2812	0.7787
R FEF activation correct trials	0.2924	0.4442	0.6571
L putamen activation correct trials	0.5367	0.9830	0.3263
R putamen activation correct trials	0.3923	0.7509	0.4532
L pPC activation correct trials	-0.6196	-0.9154	0.3606
R pPC activation correct trials	-0.8914	-1.2740	0.2035
L dlPFC activation correct trials	0.0617	0.1728	0.8628
R dlPFC activation correct trials	0.0925	0.1932	0.8468
L vlPFC activation correct trials	-0.6830	-1.3485	0.1784
R vlPFC activation correct trials	0.6435	1.3760	0.1697
dACC activation correct trials	-0.6099	-0.9989	0.3185
dACC activation error-corrected trials	1.6726	2.4183	**0.0161**

Table 2: Global significance of maternal education for each outcome measure, controlling for age

Outcome measure	df	F	p-value
Proportion correct trials	2	0.2916	0.7472
Proportion error-corrected trials	2	0.3686	0.6919
Average latency correct trials	2	2.2412	0.1080
Average latency error-corrected trials	2	0.7669	0.4653
SEF activation correct trials	2	0.0813	0.9219
Pre-SMA activation correct trials	2	0.2180	0.8042
L FEF activation correct trials	2	0.5621	0.5705
R FEF activation correct trials	2	0.4468	0.6400
L putamen activation correct trials	2	0.4275	0.6525
R putamen activation correct trials	2	0.2743	0.7602
L pPC activation correct trials	2	0.0316	0.9688
R pPC activation correct trials	2	0.4523	0.6365
L dlPFC activation correct trials	2	0.8012	0.4497
R dlPFC activation correct trials	2	0.7833	0.4577
L vlPFC activation correct trials	2	0.1464	0.8638
R vlPFC activation correct trials	2	1.0525	0.3502
dACC activation correct trials	2	2.4872	0.0847
dACC activation error-corrected trials	2	1.0098	0.3654

Table 3: Significance of smooth term for age for each final GAMM

Outcome measure	edf	F	p-value	q-value
Proportion correct trials	3.64	24.43	< 0.0001	$<$ **0.0001**
Proportion error-corrected trials	3.87	17.22	< 0.0001	$<$ **0.0001**
Average latency correct trials	2.91	12.94	< 0.0001	$<$ **0.0001**
Average latency error-corrected trials	4.02	2.29	0.0550	0.1046
SEF activation correct trials	1.00	0.00	0.9644	0.9828
Pre-SMA activation correct trials	1.00	0.97	0.3244	0.4867
L FEF activation correct trials	2.67	4.44	0.0142	**0.0320**
R FEF activation correct trials	1.00	3.11	0.0785	0.1285
L putamen activation correct trials	1.00	0.45	0.5025	0.6461
R putamen activation correct trials	1.00	0.11	0.7368	0.8289
L pPC activation correct trials	2.74	4.49	0.0060	**0.0193**
R pPC activation correct trials	2.30	4.63	0.0055	**0.0193**
L dlPFC activation correct trials	1.00	0.00	0.9828	0.9828
R dlPFC activation correct trials	3.15	4.20	0.0064	**0.0193**
L vlPFC activation correct trials	1.54	0.27	0.7332	0.8289
R vlPFC activation correct trials	2.57	2.23	0.0581	0.1046
dACC activation correct trials	1.00	0.84	0.3570	0.4948
dACC activation error-corrected trials*	1.51	4.55	0.0120	**0.0309**

*Model adjusted for sex

3.2 Predicting health-related quality of life

The second objective of the present analysis was to determine how individual differences in the adolescent development of inhibitory control, as estimated using GAMMs, may predict subsequent health-related quality of life in early adulthood. Random intercepts and slopes from the final GAMM models for which there was a significant effect of age on the respective outcome measure, after controlling for the false discovery rate, were standardized and used as candidate predictors in

a bootstrap-enhanced elastic net regression model to predict the composite health-related quality of life score derived from the WHO-QOL questionnaire. Random intercepts are considered to represent person-specific deviations from the mean level of the respective outcome measure at the mean age of the sample (20.71 years), while random slopes are considered to represent person-specific deviations in the rate of development of that measure. The candidate predictors for the elastic net model included the standardized random intercepts and slopes for the proportion of correct trials on the antisaccade task, proportion of error-corrected trials on the antisaccade task, average latency on correct trials, activation (percent BOLD signal change relative to baseline) during correct antisaccade trials in L FEF, L pPC, R pPC, R dlPFC, and activation during error-corrected antisaccade trials in dACC.

Ten-fold cross validation was first used to simultaneously tune the elastic net hyperparameters, α and λ, which control the relative contributions of the ridge and lasso penalty terms, and the level of shrinkage, respectively. The hyperparameter values that minimized the cross-validated MSE (CV-MSE) were $\hat{\alpha} = 0.1$ and $\hat{\lambda} = 0.7753$, with CV-MSE = 0.9866. With these optimal hyperparameter values, the elastic net model selected 10 of the 16 candidate predictors, including random intercepts and slopes for the proportion of correct antisaccade trials, proportion of error-corrected antisaccade trials, average latency on correct antisaccade trials, and activation during correct antisaccade trials in L FEF, as well as random intercepts for activation during correct antisaccade trials in R pPC and R dLPFC. Estimated regression coefficients for the fitted model are presented in Table 4. A coefficient estimate exactly equal to zero indicates that the predictor was not selected for inclusion in the model. The model had an $R^2 = 0.1884$, indicating that only about 18.84% of the total variability in composite health-related quality of life score was explained by the elastic net model containing this set of selected covariates.

Results from the bootstrap-enhanced procedure, in which the model was refit on $B = 5000$ bootstrap samples, suggested that none of the selected covariates were, given the other covariates in the model, statistically significant predictors of composite health-related quality of life score (Table 4). Specifically, all of the 95% bootstrapped confidence intervals (CI) for the estimated regression coefficients contained zero, suggesting that none of these coefficients significantly differed from zero. Additionally, the variable inclusion probability (VIP) for each predictor, which quantifies the importance of that predictor as the proportion of times, out of the 5000 bootstrap iterations, the predictor received a non-zero coefficient estimate, were also below 95%. Although below this chosen threshold for statistical significance, several predictors had relatively high VIPs, including random

slopes for activation during correct antisaccade trials in L FEF (VIP = 0.8916) and the proportion of error-corrected antisaccade trials (VIP = 0.8200), as well as random intercepts for activation during correct antisaccade trials in R dlPFC (VIP = 0.8456), indicating that these individual predictors may be relatively more important.

Table 4: Estimated coefficients and bootstrapped 95% confidence intervals, and variable inclusion probabilities for each candidate predictor in the elastic net model

Predictor	$\hat{\beta}$	95% CI	VIP
Antisaccade proportion correct trials intercept	0.0286	(-0.0014, 0.1216)	0.6630
Antisaccade proportion correct trials slope	0.0225	(0.0000, 0.1012)	0.6216
Antisaccade proportion error-corrected trials intercept	-0.0247	(-0.1332, 0.0104)	0.6468
Antisaccade proportion error-corrected trials slope	-0.0712	(-0.1500, 0.0000)	0.8200
Antisaccade average latency correct trials intercept	0.0284	(0.0000, 0.1282)	0.6590
Antisaccade average latency correct trials slope	-0.0118	(-0.1647, 0.0305)	0.6178
L FEF activation correct trials intercept	-0.0334	(-0.1163, 0.0000)	0.7206
L FEF activation correct trials slope	0.0858	(0.0000, 0.1713)	0.8916
L pPC activation correct trials intercept	0	(-0.0606, 0.0281)	0.3616
L pPC activation correct trials slope	0	(-0.0111, 0.0738)	0.4216
R pPC activation correct trials intercept	-0.0735	(-0.1725, 0.0000)	0.7520
R pPC activation correct trials slope	0	(-0.1277, 0.0621)	0.5396
R dlPFC activation correct trials intercept	-0.0888	(-0.2028, 0.0000)	0.8456
R dlPFC activation correct trials slope	0	(-0.1264, 0.0520)	0.4602
dACC activation error-corrected trials intercept	0	(-0.0694, 0.0775)	0.4476
dACC activation error-corrected trials slope	0	(-0.1004, 0.0462)	0.4746

To better understand the potential associations between the three predictors with relatively high VIPs and health-related quality of life, a median split was used to categorize individuals into high and low health-related quality of life groups, the GAMMs for these three measures were refit including a smooth age by health-related quality of life group interaction term, and the estimated developmental trajectories for each group were plotted (Figure 6).

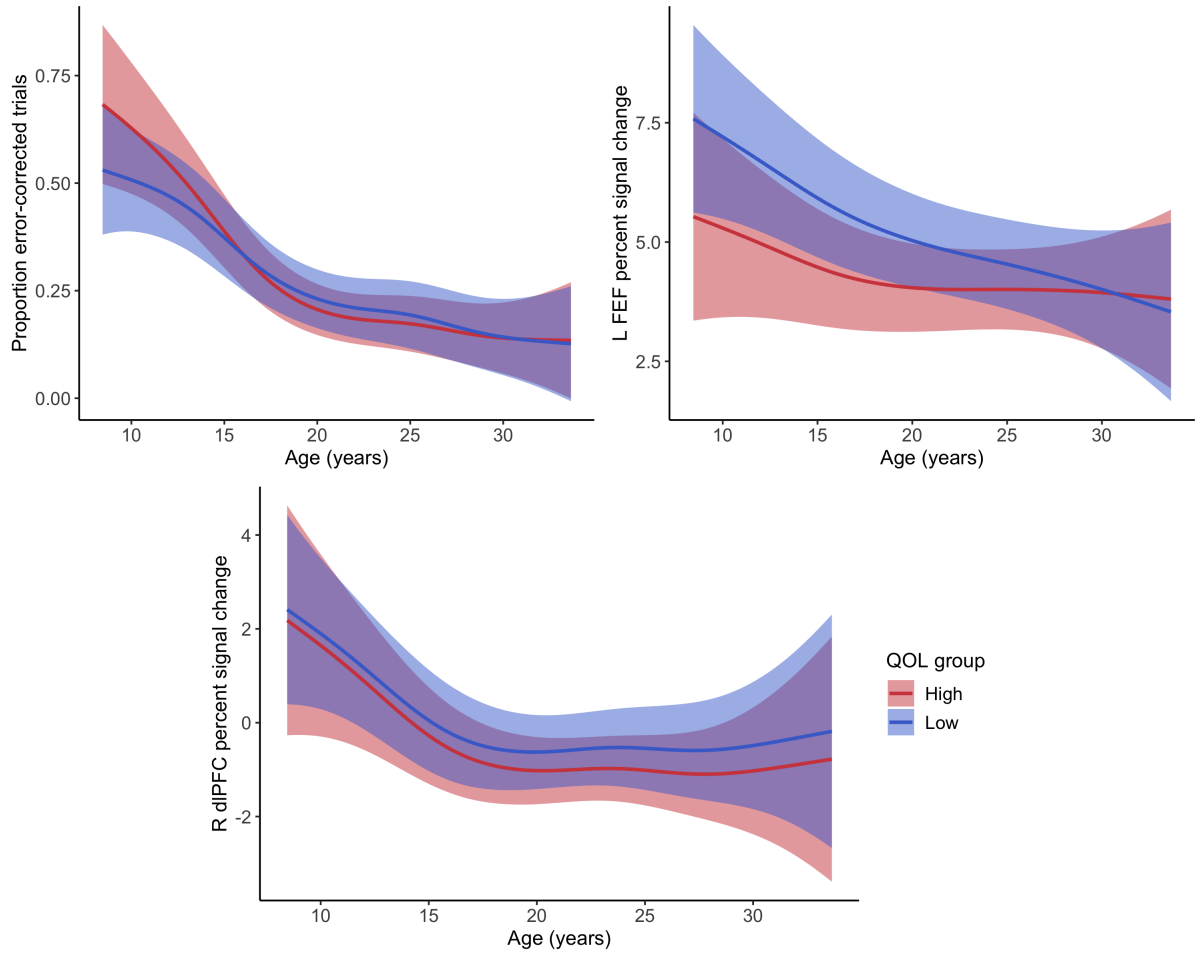

Figure 6: Estimated developmental trajectories in high (red) and low (blue) health-related quality of life (QOL) groups. Individuals with high QOL in adulthood corrected more antisaccade errors early in development than did those with low QOL in adulthood, although trajectories converged by adolescence (top left). Individuals with low QOL in adulthood had greater activation in L FEF (top right) and R dlPFC (bottom) over development compared to those with high QOL in adulthood

For the two measures for which the random slope terms were important in the elastic net model, namely, the proportion of error-corrected antisaccade trials and activation in L FEF, the plots suggest between-group differences early in development with equifinality in trajectories by adulthood. Specifically, the group with high health-related quality of life in adulthood corrected a larger proportion of antisaccade errors early in development than did those with low health-related quality of life, although these differences normalized by adolescence. Individuals with low health-

related quality of life in adulthood had greater activation in L FEF during correct antisaccade trials than did those with high quality of life, and these differences continued into young adulthood. For the measure for which the random intercept term was important in the elastic net model, activation in R dlPFC, the plot suggests parallel trajectories in both the high and low health-related quality of life groups, although activation was slightly greater across age among those with low health-related quality of life in adulthood. However, none of these smooth age by health-related quality of life group interaction effects were significant ($p \geq 0.2040$), and the bootstrap-enhanced elastic net procedure ultimately indicated that individual differences in the development of inhibitory control during adolescence were not predictive of health-related quality of life in early adulthood.

4.0 Discussion

The goals of the present study were to characterize the normative adolescent development of inhibitory control, in terms of both behavior and the function of brain regions supporting this cognitive process, and to determine how individual differences in these adolescent developmental trajectories may predict health-related quality of life in early adulthood. We examined the development of behavioral and brain function measures using GAMM, a flexible, semiparametric method that is relatively novel to functional neuroimaging research. The use of GAMMs to model the development of aspects of inhibitory control allowed us to capture nonlinear effects of age without assuming functional forms for these relationships, thus providing us with a more nuanced understanding of the shapes of developmental trajectories during adolescence and into young adulthood.

Consistent with the literature [19, 20, 22], we found significant developmental improvements in inhibitory control behaviorally, with the proportions of correctly performed and error-corrected antisaccade trials increasing and decreasing with age, respectively; average latency on correct antisaccade trials also declined with age. Post-hoc analyses revealed that developmental change in these behavioral measures continued into early adulthood, reaching mature levels as late as age 23, providing novel evidence that, behaviorally, inhibitory control may undergo a more protracted development than previously reported [20].

We also found significant effects of age for a number of brain function measures. Consistent with previous findings [22], among examined executive control regions, we found that R dlPFC activation during correct antisaccade trials significantly decreased with age. Although the majority of the examined motor response regions did not exhibit developmental changes in activation during correct antisaccade trials, significant effects of age were found for activation in motor response regions including L FEF, L pPC, and R pPC. This contrasts with previous research reporting that motor areas do not exhibit age-related changes in activation during adolescence, rather reaching mature levels of activation in childhood [22, 34]. The present finding could possibly be explained by our use of GAMMs, which may have permitted us to identify subtle age-related changes that were not evident in previous work employing linear or polynomial models, or comparisons of broad age groups. Additionally, previous findings have suggested that executive regions including the R dlPFC develop later than do motor response regions, with the former continuing into adolescence [22]. However, we found that the development of R dlPFC activation during correct antisaccade

trials occurred in parallel with that of L FEF, L pPC, and R pPC, with age-related decreases in activation in these regions beginning at approximately age 10 and continuing until approximately age 17. A previous cross-sectional study of functional connectivity associated with the antisaccade task found that the strength of long-range connections between regions in the frontal and parietal cortices increased with age [89]. Therefore, our finding of parallel developmental trajectories in such regions in the present study could reflect the integration of these regions into such a circuitry supporting inhibitory control collectively, thereby reducing processing demands on each region individually. Finally, we found that activation in dACC changed significantly with age and matured relatively later at 21.49 years, but only for error-corrected antisaccade trials, potentially reflecting its unique role in inhibitory control as a performance monitoring and and error-processing region [22, 34, 35].

Importantly, the use of a mixed model approach such as GAMM in this series of analyses allowed us to not only estimate population-level effects of age on aspects of inhibitory control, but also to estimate individual differences, which were utilized as covariates in a subsequent analysis. Specifically, we estimated random slopes, representing person-specific deviations in the rate of development through adolescence of each behavioral and brain measure, as well as random intercepts, representing person-specific deviations from the mean level of each measure at the mean age of the sample, 20.71 years. Additionally, because we found that many of the examined measures reached or were approaching maturity by this age, random intercepts also, and perhaps more meaningfully, may be thought to reflect person-specific deviations from the mean level of each measure approximately at or near the age of maturation.

These estimated individual differences were utilized as candidate predictors in a bootstrap-enhanced elastic net regression approach to predict health-related quality of life in young adulthood. The use of a regularized regression method such as elastic net was advantageous due to the relatively small size of our sample, and the potential for the number of predictors to be relatively large in comparison. Further, the bootstrap-enhanced elastic net procedure allowed us to not only perform variable selection, but also to conduct inference on the selected model, which is generally not possible with regularized regression methods. The elastic net procedure selected a model including a set of 10 of the 16 candidate predictors encompassing individual differences in adolescent development of both brain and behavioral measures. However, the model had little explanatory power, explaining only about 18.84% of the variability in the composite health-related quality of life score. This was not entirely unexpected, given that quality of life is also likely to be related

to a number of sociodemographic and situational factors [38–41] in addition to cognitive factors. Further, both the 95% bootstrap confidence intervals for the estimated coefficients and the variable inclusion probabilities indicated that, conditional on the other covariates in the model, none of the selected variables were statistically significant predictors of health-related quality of life in young adulthood. Nevertheless, an examination of the variable inclusion probabilities did suggest that three predictors in particular (random slopes for the proportion of error-corrected antisaccade trials and activation in L FEF, and random intercepts for activation in R dlPFC) may be relatively important. Post-hoc analyses indicated that those with higher health-related quality of life in adulthood corrected more antisaccade errors in late childhood and early adolescence than did those with lower health-related quality of life, although trajectories converged by mid-adolescence. This suggests that greater engagement in performance monitoring early in development may be associated with better quality of life in early adulthood. Additionally, we found that those with lower health-related quality of life in adulthood had greater activation in L FEF and R dlPFC during correctly performed antisaccade trials, potentially reflecting increased effort to perform inhibitory control as compared to individuals with higher health-related quality of life. For activation in L FEF, we observed differing trajectories between groups in late childhood and adolescence, with equifinality in adulthood, while, for activation in R dlPFC, we observed parallel developmental trajectories in the two health-related quality of life groups; this is consistent with the random slope term for L FEF activation, and the random intercept term for R dlPFC activation, being relatively more important in the elastic net model. Although these results were not statistically significant, they do motivate further investigation.

Several limitations of the present study should be considered. A large portion of our initial longitudinal sample was not eligible for inclusion in these analyses. Over time, a substantial number of participants met exclusion criteria and were dropped from the study, or were lost to follow-up and thus had fewer than three analyzable time points, were unable to be contacted to complete the WHO-QOL questionnaire at the study endpoint, or both. Additionally, among our eligible sample, the total number of analyzable observations was large ($n = 316$), but this comprised data from only 50 participants. This small sample size may have reduced our statistical power to identify significant effects, or conversely, reduced the likelihood that those results we found to be statistically significant actually reflected a true effect. Low statistical power is a common limitation in neuroscience research, and particularly for human neuroimaging studies [90]. There are also limitations associated with fMRI as a neuroimaging modality. FMRI data are inherently noisy,

with potential sources of noise including the participant's respiration, heart rate, and movement in the scanner, as well as scanner signal drifts and image artifacts [91]. Noise can result in reduced BOLD contrast sensitivity, thus compromising the ability to detect task-related changes in the BOLD signal [91]. Although we censored volumes with substantial motion and included motion parameters and baseline signal drifts as nuisance regressors in the voxelwise GLMs, we did not control for physiological recordings including respiration and heart rate. Finally, the scope of the present study was limited; in addition to behavioral measures of inhibitory control, we considered only activation in a small subset of brain regions identified from previous literature, and did not examine other regions or measures such as functional connectivity between regions.

In sum, by using a flexible, semiparametric approach that is relatively novel in the context of functional neuroimaging, we have extended important previous work by providing additional insight into the developmental time course of inhibitory control and the brain systems that support this cognitive process during adolescence. Our present findings indicate that the maturation of inhibitory control may occur later than previously reported, with refinements in behavior and brain function continuing through late adolescence and in some cases into early adulthood. These findings may have important implications for public health, as they suggest that there may be a greater window of opportunity during which interventions can be applied to effectively promote optimal adult outcomes. The relationship between the adolescent development of inhibitory control and subsequent health-related quality of life in young adulthood, however, remains to be elucidated. Although our findings did not reach the threshold for statistical significance, results did indicate that individual differences in the adolescent development of a small number of the examined brain and behavioral measures were relatively more important for predicting health-related quality of life than were others, thus motivating further work to investigate these potential associations and their implications. Future research should additionally consider examining the relationship between individual differences in the adolescent development of inhibitory control and specific domains of health-related quality of life (e.g., psychological) in early adulthood, as well as how health-related quality of life may be related to cognitive control more generally among healthy adolescents and young adults.

Appendix

R code

```
#########################################
### Score WHO-QOL questionnaire data ###
#########################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(lubridate)

## import data collected at behavioral visits
whoqol_beh <- read.csv('Data/CogLong_WHOQOL.csv',
                       header = TRUE,
                       stringsAsFactors = FALSE)

# filter empty rows, select only needed columns
whoqol_beh <- whoqol_beh %>%
  filter(!is.na(ID)) %>%
  select(-starts_with('D_')) %>%
  rename('lunaid' = 'ID') %>%
  mutate(date = as.POSIXct(mdy(date)))

## import data collected via R03 online surveys
whoqol_ro3_a <- read.csv('Data/R03+Research+Follow-up+Battery_November+14,+2019_11.43.csv',
                         header = TRUE,
                         stringsAsFactors = FALSE)

# filter unneeded rows and select only needed columns
whoqol_ro3_a <- whoqol_ro3_a[3:nrow(whoqol_ro3_a), ] %>%
  filter(Status != 'Survey Preview') %>%
  select(c('ExternalReference', 'EndDate', 'Q60_1':'Q68')) %>%
  mutate(EndDate = as.POSIXct(EndDate))

# rename columns with consistent names
names(whoqol_ro3_a) <- names(whoqol_beh)

## import data collected via R03 online surveys
whoqol_ro3_b <- read.csv('Data/R03+WHOQOL-BREF_November+15,+2019_08.48.csv',
                         header = TRUE,
                         stringsAsFactors = FALSE)

# filter unneeded rows and select only needed columns
whoqol_ro3_b <- whoqol_ro3_b[3:nrow(whoqol_ro3_b), ] %>%
  filter(Status != 'Survey Preview') %>%
  select(c('ExternalReference', 'EndDate', 'Q3_1':'Q15')) %>%
  mutate(EndDate = as.POSIXct(EndDate, format = '%m/%d/%Y %H:%M', tz = ''))

# rename columns with consistent names
```

```r
names(whoqol_ro3_b) <- names(whoqol_beh)

## merge two online survey versions
whoqol_ro3 <- rbind(whoqol_ro3_a, whoqol_ro3_b, stringsAsFactors = FALSE)

## function to recode online survey items from text input to numeric values
whoqol_text_to_num <- function(x) {
  x <- gsub(x, pattern = '^Very Poor$', replacement = '1')
  x <- gsub(x, pattern = '^Poor$', replacement = '2')
  x <- gsub(x, pattern = '^Neither [Pp]oor nor [Gg]ood$', replacement = '3')
  x <- gsub(x, pattern = '^Good$', replacement = '4')
  x <- gsub(x, pattern = '^Very [Gg]ood$', replacement = '5')

  x <- gsub(x, pattern = '^Very Dissatisfied$', replacement = '1')
  x <- gsub(x, pattern = '^Dissatisfied$', replacement = '2')
  x <- gsub(x, pattern = '^Neither [Dd]issatisfied nor [Ss]atisfied$', replacement =
      '3')
  x <- gsub(x, pattern = '^Satisfied$', replacement = '4')
  x <- gsub(x, pattern = '^Very Satisfied$', replacement = '5')

  x <- gsub(x, pattern = '^Not at all$', replacement = '1')
  x <- gsub(x, pattern = '^A little$', replacement = '2')
  x <- gsub(x, pattern = '^A moderate amount$', replacement = '3')
  x <- gsub(x, pattern = '^Very much$', replacement = '4')
  x <- gsub(x, pattern = '^An extreme amount$', replacement = '5')

  x <- gsub(x, pattern = '^Extremely$', replacement = '5')

  x <- gsub(x, pattern = '^Moderately$', replacement = '3')
  x <- gsub(x, pattern = '^Mostly$', replacement = '4')
  x <- gsub(x, pattern = '^Completely$', replacement = '5')

  x <- gsub(x, pattern = '^Never$', replacement = '5')
  x <- gsub(x, pattern = '^Seldom$', replacement = '4')
  x <- gsub(x, pattern = '^Quite Often$', replacement = '3')
  x <- gsub(x, pattern = '^Very Often$', replacement = '2')
  x <- gsub(x, pattern = '^Always$', replacement = '1')

  x <- as.numeric(x)
  return(x)
}

# recode online survey items from text input to numeric values
whoqol_ro3[, 3:28] <- sapply(whoqol_ro3[, 3:28], whoqol_text_to_num)

# items 3 and 4 are reverse scored
whoqol_ro3$X3 <- 6 - whoqol_ro3$X3
whoqol_ro3$X4 <- 6 - whoqol_ro3$X4

# replace missing value with mean of other domain items per scoring guidelines
which(is.na(whoqol_ro3), arr.ind = TRUE)
whoqol_ro3[19, 17] <- mean(unlist(whoqol_ro3[19, c('X3', 'X4', 'X10', 'X16', 'X17',
    'X18')]))

## merge online survey data with behavioral visit data
whoqol <- rbind(whoqol_ro3, whoqol_beh, stringsAsFactors = FALSE)

## score whoqol per scoring guidelines
whoqol <- whoqol %>%
```

```r
  mutate(D_1_Raw = X3 + X4 + X10 + X15 + X16 + X17 + X18) %>% # physical
  mutate(D_2_Raw = X5 + X6 + X7 + X11 + X19 + X26) %>% # psychological
  mutate(D_3_Raw = X20 + X21 + X22) %>% # social
  mutate(D_4_Raw = X8 + X9 + X12 + X13 + X14 + X23 + X24 + X25) # environment

## functions to rescale raw scores
scale_domain_1 <- function(x) {
  y <- case_when(x == 7 ~ 4,
                 x == 8 | x == 9 ~ 5,
                 x == 10 | x == 11 ~ 6,
                 x == 12 | x == 13 ~ 7,
                 x == 14 ~ 8,
                 x == 15 | x == 16 ~ 9,
                 x == 17 | x == 18 ~ 10,
                 x == 19 | x == 20 ~ 11,
                 x == 21 ~ 12,
                 x == 22 | x == 23 ~ 13,
                 x == 24 | x == 25 ~ 14,
                 x == 26 | x == 27 ~ 15,
                 x == 28 ~ 16,
                 x == 29 | x == 30 ~ 17,
                 x == 31 | x == 32 ~ 18,
                 x == 33 | x == 34 ~ 19,
                 x == 35 ~ 20)
  return(y)
}

scale_domain_2 <- function(x) {
  y <- case_when(x == 6 ~ 4,
                 x == 7 | x == 8 ~ 5,
                 x == 9 ~ 6,
                 x == 10 | x == 11 ~ 7,
                 x == 12 ~ 8,
                 x == 13 | x == 14 ~ 9,
                 x == 15 ~ 10,
                 x == 16 | x == 17 ~ 11,
                 x == 18 ~ 12,
                 x == 19 | x == 20 ~ 13,
                 x == 21 ~ 14,
                 x == 22 | x == 23 ~ 15,
                 x == 24 ~ 16,
                 x == 25 | x == 26 ~ 17,
                 x == 27 ~ 18,
                 x == 28 | x == 29 ~ 19,
                 x == 30 ~ 20)
  return(y)
}

scale_domain_3 <- function(x) {
  y <- case_when(x == 3 ~ 4,
                 x == 4 ~ 5,
                 x == 5 ~ 7,
                 x == 6 ~ 8,
                 x == 7 ~ 9,
                 x == 8 ~ 11,
                 x == 9 ~ 12,
                 x == 10 ~ 13,
                 x == 11 ~ 15,
                 x == 12 ~ 16,
```

```
                    x == 13 ~ 17,
                    x == 14 ~ 19,
                    x == 15 ~ 20)
  return(y)
}

scale_domain_4 <- function(x) {
  y <- case_when(x == 8 ~ 4,
                 x == 9 | x == 10 ~ 5,
                 x == 11 | x == 12 ~ 6,
                 x == 13 | x == 14 ~ 7,
                 x == 15 | x == 16 ~ 8,
                 x == 17 | x == 18 ~ 9,
                 x == 19 | x == 20 ~ 10,
                 x == 21 | x == 22 ~ 11,
                 x == 23 | x == 24 ~ 12,
                 x == 25 | x == 26 ~ 13,
                 x == 27 | x == 28 ~ 14,
                 x == 29 | x == 30 ~ 15,
                 x == 31 | x == 32 ~ 16,
                 x == 33 | x == 34 ~ 17,
                 x == 35 | x == 36 ~ 18,
                 x == 37 | x == 38 ~ 19,
                 x == 39 | x == 40 ~ 20)
  return(y)
}

scale_20_to_100 <- function(x) {
  y <- case_when(x == 4 ~ 0,
                 x == 5 ~ 6,
                 x == 6 ~ 13,
                 x == 7 ~ 19,
                 x == 8 ~ 25,
                 x == 9 ~ 31,
                 x == 10 ~ 38,
                 x == 11 ~ 44,
                 x == 12 ~ 50,
                 x == 13 ~ 56,
                 x == 14 ~ 63,
                 x == 15 ~ 69,
                 x == 16 ~ 75,
                 x == 17 ~ 81,
                 x == 18 ~ 88,
                 x == 19 ~ 94,
                 x == 20 ~ 100)
  return(y)
}

# transform raw domain scores to scaled scores
whoqol <- whoqol %>%
  mutate(D_1_Trans_4_20 = scale_domain_1(D_1_Raw)) %>%
  mutate(D_2_Trans_4_20 = scale_domain_2(D_2_Raw)) %>%
  mutate(D_3_Trans_4_20 = scale_domain_3(D_3_Raw)) %>%
  mutate(D_4_Trans_4_20 = scale_domain_4(D_4_Raw)) %>%
  mutate(D_1_Trans_0_100 = scale_20_to_100(D_1_Trans_4_20)) %>%
  mutate(D_2_Trans_0_100 = scale_20_to_100(D_2_Trans_4_20)) %>%
  mutate(D_3_Trans_0_100 = scale_20_to_100(D_3_Trans_4_20)) %>%
  mutate(D_4_Trans_0_100 = scale_20_to_100(D_4_Trans_4_20))

```

```

## some subjects completed the questionnaire more than once
# create counter variable to keep track of repeated administrations
whoqol <- whoqol %>%
  group_by(lunaid) %>%
  arrange(date, .by_group = TRUE) %>%
  mutate(counter = row_number())

# select only scores from subject's most recent administration
whoqol_most_recent <- whoqol %>%
  group_by(lunaid) %>%
  filter(counter == max(counter))

write.csv(whoqol_most_recent, 'Data/whoqol_ages_scores_20200204.csv')


## descriptive stats
whoqol_variability <- data.frame(matrix(NA, nrow = 1, ncol = 6))
whoqol_variability <- whoqol_variability[-1, , drop = FALSE]
names(whoqol_variability) <- c('Measure', 'Mean', 'SD', 'Median', 'Min', 'Max')
columns <- names(whoqol_most_recent)[30:41]

for (i in seq(1, length(columns), by = 1)){
  whoqol_variability[i, 'Measure'] <- columns[i]
  whoqol_variability[i, 'Mean'] <- mean(as.numeric(unlist(whoqol_most_recent[,
      columns[i]])))
  whoqol_variability[i, 'SD'] <- sd(as.numeric(unlist(whoqol_most_recent[, columns[i
      ]])))
  whoqol_variability[i, 'Median'] <- median(as.numeric(unlist(whoqol_most_recent[,
      columns[i]])))
  whoqol_variability[i, 'Min'] <- range(as.numeric(unlist(whoqol_most_recent[,
      columns[i]])))[1]
  whoqol_variability[i, 'Max'] <- range(as.numeric(unlist(whoqol_most_recent[,
      columns[i]])))[2]
}
```

```
#########################################################################
### Plot of final sample age distribution and descriptive stats ###
#########################################################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)

# cog subject DOB and sex pulled from database
dob_sex <- read.csv('Data/cog_allsubs_dobs_sex.csv',
                    header = TRUE,
                    stringsAsFactors = FALSE)
dob_sex <- dob_sex %>%
  rename('lunaid' = 'id') %>%
  mutate(dob = as.Date(as.character(dob), format = '%Y-%m-%d'))


# scored whoqol data compiled with prev script
whoqol <- read.csv('Data/whoqol_ages_scores_20200204.csv',
```

```
                    header = TRUE,
                    stringsAsFactors = FALSE)
whoqol <- whoqol %>%
  select(-c('X')) %>%
  mutate(date = as.Date(as.character(date), format = '%Y-%m-%d'))

# join dob and whoqol data
whoqol_age <- whoqol %>%
  left_join(dob_sex, by = 'lunaid') %>%
  mutate(age = interval(start = dob, end = date)/duration(n = 1, unit = 'years')) %>%
  mutate(age_type = 'WHO-QOL') %>% # indicator if age is for scan or for whoqol
  mutate(note = '') %>%
  select(c(lunaid, date, dob, sex, age, note, age_type)) %>%
  rename('vtimestamp' = 'date')

# subjects who completed whoqol
subids <- unique(whoqol_age$lunaid)
length(subids)


# Cog scan dates and ages pulled from database
scan_age <- read.csv('Data/cog_scans_dates_ages.csv',
                     header = TRUE,
                     stringsAsFactors = FALSE)
scan_age <- scan_age %>%
  mutate(dob = as.Date(as.character(dob), format = '%Y-%m-%d')) %>%
  mutate(vtimestamp = as.Date(as.character(vtimestamp), format = '%Y-%m-%d')) %>%
  rename('lunaid' = 'id') %>%
  mutate(age_type = 'Scan') %>% # indicator if age is for scan or for whoqol
  mutate(visit_identifier = paste(lunaid, vtimestamp, sep = '_'))

# remove dropped visits
scan_age_nodrop <- scan_age %>%
  filter(lunaid %in% subids) %>% # select only subjects who completed whoqol
  filter(vtimestamp > '2005-11-11') %>% # dates prior to Nov 2005 = pilot data
  filter(visit_identifier != '10188_2005-12-14') %>%
  filter(visit_identifier != '10192_2005-12-22') %>%
  filter(visit_identifier != '10463_2007-10-24') %>%
  filter(visit_identifier != '10463_2008-11-08') %>%
  filter(visit_identifier != '10365_2017-05-23') %>%
  filter(visit_identifier != '10133_2009-01-14') %>%
  filter(visit_identifier != '10248_2006-06-26') %>%
  filter(visit_identifier != '10136_2007-07-21') %>%
  filter(visit_identifier != '10132_2008-12-23') %>%
  filter(visit_identifier != '10184_2010-09-25') %>%
  filter(visit_identifier != '10252_2011-02-09') %>%
  filter(visit_identifier != '10202_2009-01-06') %>%
  select(-c('visit_identifier'))

# sessions to preprocess scan data for
write.csv(scan_age_nodrop, 'Data/cog_subs_to_preproc_20200217.csv')


## final sample age distribution
# subjects who have whoqol data and 3+ usable/preprocessed scan time points
final_subs <- read.csv('Data/final_subs_for_analysis_20200228.csv',
                       stringsAsFactors = FALSE)
final_subs <- final_subs %>%
```

```
  select(-c(X, n_scans)) %>%
  mutate(date = as.Date(date, format = '%Y-%m-%d')) %>%
  rename('lunaid' = 'id') %>%
  rename('vtimestamp' = 'date')

final_subids <- unique(final_subs$lunaid)

# merge whoqol, scan age, and final sample data
dat <- final_subs %>%
  left_join(scan_age_nodrop,  by = c('lunaid', 'vtimestamp')) %>%
  mutate(visit_identifier = paste(lunaid, vtimestamp, sep = '_')) %>%
  filter(!duplicated(visit_identifier)) %>%
  select(-c('visit_identifier')) %>%
  bind_rows(whoqol_age) %>%
  filter(lunaid %in% final_subids)

write.csv(dat, 'Data/whoqol_and_scan_ages_20200228.csv')

# number of scans per subject
numscans <- as.data.frame(table(dat$lunaid) - 1) # one observation is for whoqol
colnames(numscans)[1] <- 'lunaid'

# order subjects by age at baseline scan
min_ages <- dat %>%
  filter(age_type == 'Scan') %>%
  select(lunaid, age) %>%
  group_by(lunaid) %>%
  filter(age == min(age))
min_ages <- min_ages[order(min_ages$age), ]
min_ages$order_for_plot <- seq(1:nrow(min_ages))
range(min_ages$age)

# data for plotting
dat_to_plot <- min_ages %>%
  select(lunaid, order_for_plot) %>%
  full_join(dat, by = 'lunaid')

# data for plotting - scans only
dat_to_plot_nowhoqol <- dat_to_plot %>%
  filter(age_type != 'WHO-QOL')

# create plot of longitudinal subject age distribution
subject_plot <- ggplot(dat_to_plot_nowhoqol) +
  aes(y = as.factor(order_for_plot),
      x = age,
      color = factor(sex, labels = c('Female', 'Male')),
      group = as.factor(order_for_plot))  +
  geom_line() +
  geom_point(data = dat_to_plot,
             aes(shape = as.factor(age_type))) +
  scale_colour_manual(values = c('#D1474B', '#476FD1')) +
  theme_bw() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), axis.line = element_line(colour = "black
          ")) +
  labs(x = 'Age (years)', y = 'Subject', color = 'Sex', shape = 'Visit type')
subject_plot

ggsave('subjects_20200228.png', plot = subject_plot, width = 5, height = 6, units =
```

```
    'in')

## descriptive statistics
# create variables for intervals (years) between each scan timepoint
bw_scan_intervals <- dat_to_plot_nowhoqol %>%
  select(c(lunaid, vtimestamp)) %>%
  group_by(lunaid) %>%
  arrange(vtimestamp, .by_group = TRUE) %>%
  mutate(counter = row_number()) %>%
  pivot_wider(id_cols = lunaid, names_from = counter, values_from = vtimestamp) %>%
  mutate(interval1 = interval(start = `1`, end = `2`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval2 = interval(start = `2`, end = `3`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval3 = interval(start = `3`, end = `4`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval4 = interval(start = `4`, end = `5`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval5 = interval(start = `5`, end = `6`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval6 = interval(start = `6`, end = `7`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval7 = interval(start = `7`, end = `8`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval8 = interval(start = `8`, end = `9`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval9 = interval(start = `9`, end = `10`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval10 = interval(start = `10`, end = `11`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval11 = interval(start = `11`, end = `12`)/duration(n = 1, unit = 'years')) %>%
  mutate(interval12 = interval(start = `12`, end = `13`)/duration(n = 1, unit = 'years')) %>%
  select(-c(`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`)) %>%
  pivot_longer(cols = starts_with('interval'),
               names_to = 'bw_scan_interval_length')

# average interval between scans
mean(as.numeric(bw_scan_intervals$value), na.rm = TRUE)
sd(as.numeric(bw_scan_intervals$value), na.rm = TRUE)
min(as.numeric(bw_scan_intervals$value), na.rm = TRUE)
max(as.numeric(bw_scan_intervals$value), na.rm = TRUE)

# average number of scans among the final sample (those with 3+ useable time points)
mean(as.numeric(numscans$Freq))
sd(as.numeric(numscans$Freq))
min(as.numeric(numscans$Freq))
max(as.numeric(numscans$Freq))

# average baseline age (age at first scan)
mean(as.numeric(min_ages$age))
sd(as.numeric(min_ages$age))
min(as.numeric(min_ages$age))
max(as.numeric(min_ages$age))

# average endpoint age (age at final scan)
max_ages <- dat %>%
```

```
  filter(age_type == 'Scan') %>%
  select(lunaid, age) %>%
  group_by(lunaid) %>%
  filter(age == max(age))
mean(as.numeric(max_ages$age))
sd(as.numeric(max_ages$age))
min(as.numeric(max_ages$age))
max(as.numeric(max_ages$age))

# n and proportion male and female
whoqol_age %>%
  filter(lunaid %in% final_subids) %>%
  group_by(sex) %>%
  summarize(n = n(),
            prop = n()/length(final_subids))
```

```
##############################################
### Score antisaccade eye-tracking data ###
##############################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(lubridate)

# autoeyescored eye data
eye_data <- read.table('Data/all_trail_et.tsv',
                       sep = '\t',
                       header = TRUE,
                       stringsAsFactors = FALSE)

eye_data <- eye_data %>%
  mutate(date = as.Date(as.character(date), format = '%Y%m%d')) %>%
  mutate(fstCorrect <- as.numeric(fstCorrect)) %>%
  mutate(ErrCorr <- as.numeric(ErrCorr)) %>%
  mutate(Incorrect = ifelse(Count == 0, 1, 0)) %>%
  mutate(Dropped = ifelse(Count == -1, 1, 0))

## accuracy
# proportion of each trial type in each session (considering all 4 anti runs
    together)
anti_perf <- eye_data %>%
  filter(AS == 'AS') %>%
  group_by(id, date) %>%
  summarise(anti_perc_correct = mean(fstCorrect),
            anti_perc_error_corrected = mean(ErrCorr),
            anti_perc_incorrect = mean(Incorrect),
            anti_perc_dropped_trials = mean(Dropped))

# proportion of each trial type in each session among non-dropped trials
anti_perf_nodrop <- eye_data %>%
  filter(AS == 'AS') %>%
  filter(Dropped == 0) %>%
  group_by(id, date) %>%
  summarise(anti_perc_correct_nodrop = mean(fstCorrect),
            anti_perc_error_corrected_nodrop = mean(ErrCorr))

## saccade latency
```

```r
# average antisaccade latency on correct trials only
anti_lat_corr <- eye_data %>%
  filter(AS == 'AS') %>%
  filter(Count == 1) %>%
  group_by(id, date) %>%
  summarise(anti_avg_lat_correct_trials = mean(lat))

# average antisaccade latency on error corrected trials
anti_lat_errcorr <- eye_data %>%
  filter(AS == 'AS') %>%
  filter(Count == 2) %>%
  group_by(id, date) %>%
  summarise(anti_avg_lat_error_corrected_trials = mean(lat))


# join accuracy and latency data
anti_perf_lat_20200228 <- anti_perf %>%
  full_join(anti_perf_nodrop, by = c('id', 'date')) %>%
  full_join(anti_lat_corr, by = c('id', 'date')) %>%
  full_join(anti_lat_errcorr, by = c('id', 'date')) %>%
  rename('lunaid' = 'id')

write.csv(anti_perf_lat_20200228, 'Data/anti_perf_lat_20200228.csv')
```

```r
#####################################
### Merge WHO-QOL and eye data ###
#####################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)

# whoqol data
whoqol <- read.csv('Data/whoqol_and_scan_ages_20200228.csv',
                   header = TRUE,
                   stringsAsFactors = FALSE)

whoqol <- whoqol %>%
  select(-c('X')) %>%
  mutate(dob = as.Date(as.character(dob), format = '%Y-%m-%d')) %>%
  mutate(vtimestamp = as.Date(as.character(vtimestamp), format = '%Y-%m-%d')) %>%
  rename('date' = 'vtimestamp')

# import scored anti task eye data
eye <- read.csv('Data/anti_perf_lat_20200228.csv',
                header = TRUE,
                stringsAsFactors = FALSE)

eye <- eye %>%
  select(-c('X')) %>%
  mutate(date = as.Date(as.character(date), format = '%Y-%m-%d')) %>%
  rename('lunaid' = 'id')

# join whoqol and eye data
whoqol_plus_eye <- whoqol %>%
  left_join(eye, by = c('lunaid', 'date')) %>%
  write.csv('Data/whoqol_plus_eye_20200228.csv')
```

```
#####################################################
### Create stim time files for AFNI 3dDeconvolve ###
#####################################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(readxl)
library(dplyr)
library(tidyr)
library(LNCDR)
library(stringr)

# empty dataframe
antistate_timings <- data.frame(FRAME = integer(),
                        `Start Tm` = double(),
                        `End Tm` = double(),
                        Event = character(),
                        Location = integer(),
                        Run = character(),
                        stringsAsFactors = FALSE)

# read in relevant columns from each sheet of Katerina's task timing Excel sheet and
    append to the dataframe
for (i in 1:12){
  av <- readxl::read_excel('Data/Anti_Mix_Design_Lists_FINAL_minusfix_fixblockrecord
      _FINMOD.xls',
                           sheet = sprintf('List%sAV', i),
                           range = cell_cols('BB:BF'))
  av$Run <- sprintf('%sAV', i)
  va <- readxl::read_excel('Data/Anti_Mix_Design_Lists_FINAL_minusfix_fixblockrecord
      _FINMOD.xls',
                           sheet = sprintf('List%sVA', i),
                           range = cell_cols('BB:BF'))
  va$Run <- sprintf('%sVA', i)
  av_va <- rbind(av, va)
  antistate_timings <- rbind(antistate_timings, av_va)
}

write.csv(antistate_timings, 'Data/antistate_timings_from_kat.csv')

# subject-date run orders
sub_runs <- readxl::read_excel('Data/Anti-State&MGSEncode_data.xls',
                               sheet = 'Anti-State_sorted_by_year')
sub_runs$Scan_Date <- as.Date(sub_runs$Scan_Date)

# cleaning subject, date, run data
sub_runs <- sub_runs %>%
  select(id = Oxford_ID,
         date = Scan_Date,
         bircid = BIRC_ID,
         AS1 = `Anti-State_1`,
         AS2 = `Anti-State_2`,
         AS3 = `Anti-State_3`,
         AS4 = `Anti-State_4`) %>%
  pivot_longer(cols = AS1:AS4,
               names_to = 'run',
               values_to = 'run_type')
```

```
55
56 sub_runs$run <- str_replace_all(sub_runs$run, "^AS", "")
57 sub_runs$run <- as.integer(sub_runs$run)
58 sub_runs$run_type <- str_replace_all(sub_runs$run_type, "^.\\(", "")
59 sub_runs$run_type <- str_replace_all(sub_runs$run_type, "^. \\(", "")
60 sub_runs$run_type <- str_replace_all(sub_runs$run_type, "^..\\(", "")
61 sub_runs$run_type <- str_replace_all(sub_runs$run_type, "\\)$", "")
62
63 # scored eye data (per trial, all subjects)
64 eye_data <- read.csv('Data/all_trail_et.tsv',
65                      sep = '\t',
66                      stringsAsFactors = FALSE)
67
68 eye_data <- eye_data %>%
69   mutate(date = as.Date(as.character(date), format = '%Y%m%d')) %>%
70   mutate(fstCorrect = as.numeric(fstCorrect)) %>%
71   mutate(ErrCorr = as.numeric(ErrCorr)) %>%
72   mutate(Incorrect = ifelse(Count == 0, 1, 0)) %>%
73   mutate(Dropped = ifelse(Count == -1, 1, 0))
74
75 eye_data_sub_runs <- eye_data %>%
76   left_join(sub_runs, by = c('id', 'date', 'run'))
77 write.csv(eye_data_sub_runs, 'Data/antistate_scored_eyd_plus_run_numbers.csv')
78
79 eye_data_sub_runs  %>%
80   filter(is.na(run_type)) %>%
81   select(id, date, run, run_type) %>%
82   tally()
83
84 # subjects for analysis (have whoqol and 3+ scans)
85 subs_for_analysis <- read.csv('Data/cog_subs_to_preproc_20200217.csv',
86                               stringsAsFactors = FALSE)
87
88 subs_for_analysis <- subs_for_analysis %>%
89   mutate(vtimestamp = as.Date(vtimestamp, format = '%m/%d/%Y'))
90 names(subs_for_analysis) <- c('id', 'date', 'age', 'note')
91
92 eye_data_sub_runs_analysis <- subs_for_analysis %>%
93   left_join(eye_data_sub_runs, by = c('id', 'date'))
94
95 # filter trials that are fixation, merge for saving 1D files later
96 antistate_timings_excl_fix <- antistate_timings %>%
97   filter(!grepl('FIX$', Event)) %>% # including 'FIXCUE'
98   group_by(run_type) %>%
99   filter(Event != lag(Event) | is.na(lag(Event))) %>%
100  mutate(trial = cumsum(grepl('(ANTI|VGS)CUE', Event)))
101
102 write.csv(antistate_timings_excl_fix, 'Data/antistate_timings_from_kat_excl_fix.csv'
      )
103
104 merged_for_1d <- eye_data_sub_runs_analysis %>%
105   full_join(antistate_timings_excl_fix, by = c('trial', 'run_type'))
106
107 write.csv(merged_for_1d, 'Data/cog_antistate_timings_merged_for_1d_20200219.csv')
108
109 # create new column in merged file to specify left/right
110 unique(merged_for_1d$Location)
111 merged_for_1d <- merged_for_1d %>%
112   mutate(LeftRight = case_when(Location < 320 ~ 'Left',
```

```
                                  Location > 320 ~ 'Right'))

# write 1D files for 3Ddeconvolve
oned_return <- function(f, ...){
  save1D(fname = f, ...)
  return(f)
}

merged_for_1d <- merged_for_1d %>%
  filter(!is.na(id))

for(subid in unique(merged_for_1d$id)){
  temp <- merged_for_1d %>%
    filter(id == subid)

  for(subdate in unique(temp$date)) {
    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Left') %>%
      filter(Event == 'ANTICUE') %>%
      filter(Count == 1) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_anti_left_correct.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))

    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Left') %>%
      filter(Event == 'ANTICUE') %>%
      filter(Count == 2) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_anti_left_errcorr.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))


    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Left') %>%
      filter(Event == 'ANTICUE') %>%
      filter(Count == 0) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_anti_left_incorr.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))


    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Left') %>%
      filter(Event == 'ANTICUE') %>%
      filter(Count == -1) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_anti_left_dropped.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))

    temp %>%
```

```
172         rename(block = run) %>%
173         filter(LeftRight == 'Right') %>%
174         filter(Event == 'ANTICUE') %>%
175         filter(Count == 1) %>%
176         filter(date == subdate) %>%
177         summarize(d = strftime(first(date), format = '%Y%m%d'),
178                   f = sprintf('1D_files/%s_%s_anti_right_correct.1D', first(id), d),
179                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
180
181       temp %>%
182         rename(block = run) %>%
183         filter(LeftRight == 'Right') %>%
184         filter(Event == 'ANTICUE') %>%
185         filter(Count == 2) %>%
186         filter(date == subdate) %>%
187         summarize(d = strftime(first(date), format = '%Y%m%d'),
188                   f = sprintf('1D_files/%s_%s_anti_right_errcorr.1D', first(id), d),
189                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
190
191       temp %>%
192         rename(block = run) %>%
193         filter(LeftRight == 'Right') %>%
194         filter(Event == 'ANTICUE') %>%
195         filter(Count == 0) %>%
196         filter(date == subdate) %>%
197         summarize(d = strftime(first(date), format = '%Y%m%d'),
198                   f = sprintf('1D_files/%s_%s_anti_right_incorr.1D', first(id), d),
199                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
200
201       temp %>%
202         rename(block = run) %>%
203         filter(LeftRight == 'Right') %>%
204         filter(Event == 'ANTICUE') %>%
205         filter(Count == -1) %>%
206         filter(date == subdate) %>%
207         summarize(d = strftime(first(date), format = '%Y%m%d'),
208                   f = sprintf('1D_files/%s_%s_anti_right_dropped.1D', first(id), d),
209                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
210
211       temp %>%
212         rename(block = run) %>%
213         filter(LeftRight == 'Left') %>%
214         filter(Event == 'VGSCUE') %>%
215         filter(Count != -1) %>%
216         filter(date == subdate) %>%
217         summarize(d = strftime(first(date), format = '%Y%m%d'),
218                   f = sprintf('1D_files/%s_%s_vgs_left.1D', first(id), d),
219                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
220
221       temp %>%
222         rename(block = run) %>%
223         filter(LeftRight == 'Left') %>%
224         filter(Event == 'VGSCUE') %>%
225         filter(Count == -1) %>%
226         filter(date == subdate) %>%
227         summarize(d = strftime(first(date), format = '%Y%m%d'),
228                   f = sprintf('1D_files/%s_%s_vgs_left_dropped.1D', first(id), d),
229                   r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))
230
```

```
    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Right') %>%
      filter(Event == 'VGSCUE') %>%
      filter(Count != -1) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_vgs_right.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))

    temp %>%
      rename(block = run) %>%
      filter(LeftRight == 'Right') %>%
      filter(Event == 'VGSCUE') %>%
      filter(Count == -1) %>%
      filter(date == subdate) %>%
      summarize(d = strftime(first(date), format = '%Y%m%d'),
                f = sprintf('1D_files/%s_%s_vgs_right_dropped.1D', first(id), d),
                r = oned_return(f, ., colname = 'Start Tm', nblocks = 4))

  }
}
```

```
###################################
### Average activation in ROIs ###
###################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(tidyr)
library(dplyr)
library(stringr)
library(lubridate)

# roi names and coordinates from Ordaz et al. 2013 paper
roi_names <- read.csv('Data/sarah_rois_20200225.csv',
                      header = TRUE,
                      stringsAsFactors = FALSE)

# average activation per roi per session - output from AFNI 3dROIstats
roi_vals <- read.table('Data/jen_roi_stats_20200228.txt',
                       header = TRUE,
                       stringsAsFactors = FALSE)

# rename AFNI sub-brick labels to be more descriptive
roi_vals <- roi_vals %>%
  mutate(condition = case_when(Sub.brick == '0[anti_corr]' ~ 'anti_corr_v_baseline',
                               Sub.brick == '1[anti_errc]' ~ 'anti_errcorr_v_
                                   baseline',
                               Sub.brick == '2[anti_inco]' ~ 'anti_incorr_v_baseline
                                   ',
                               Sub.brick == '3[anti_corr]' ~ 'anti_corr_v_errcorr',
                               Sub.brick == '4[anti_corr]' ~ 'anti_corr_v_incorr',
                               Sub.brick == '5[anti_corr]' ~ 'anti_corr_v_vgs')) %>%
  mutate(File2 = str_replace_all(File, '^../subjs/...../', '')) %>%
  mutate(File2 = str_replace_all(File2, '_bucket.nii.gz.*', '')) %>%
  separate(col = File2,
           into = c('lunaid', 'date'),
```

```
            sep = '_') %>%
  select(lunaid, date, condition, everything()) %>%
  select(-c('File', 'Sub.brick')) %>%
  mutate(date = as.Date(date, format = '%Y%m%d'))

# rename columns
colnames(roi_vals) <- c('id', 'date', 'condition',
                        'SEF', 'pre_SMA', 'L_FEF', 'R_FEF',
                        'L_putamen', 'R_putamen', 'L_pPC', 'R_pPC',
                        'L_dlPFC', 'R_dlPFC', 'L_vlPFC', 'R_vlPFC',
                        'dACC')

# pivot dataframe from long to wide format
roi_vals_wide <- roi_vals %>%
  pivot_wider(id_cols = c('id', 'date'),
              names_from = c('condition'),
              values_from = c('SEF', 'pre_SMA', 'L_FEF', 'R_FEF',
                              'L_putamen', 'R_putamen', 'L_pPC', 'R_pPC',
                              'L_dlPFC', 'R_dlPFC', 'L_vlPFC', 'R_vlPFC',
                              'dACC'))

roi_vals_wide %>% write.csv('Data/roi_percent_signal_change_20200228.csv')
```

```
###################################
### Maternal education level ###
###################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(tidyr)

# demographic data for Cog subjects pulled from LNCD database
demos <- read.csv('Data/Cog_Demos_20190815.csv',
                  stringsAsFactors = FALSE)
colnames(demos)[1] <- 'lunaid'
demos$vtimestamp <- as.Date(demos$vtimestamp,
                            format = '%Y-%m-%d')

# maternal edu level at baseline visit
mat_edu <- demos %>%
  filter(level_edu_mother != 'null') %>%
  filter(level_edu_mother != '-9') %>%
  group_by(lunaid) %>%
  filter(vtimestamp == min(vtimestamp)) %>%
  select(lunaid, level_edu_mother)

# convert numeric labels in db to corresponding labels on demo form
mat_edu <- mat_edu %>%
  mutate(level_edu_mother_cat =
           case_when(level_edu_mother == 1 ~ 'Less than high school',
                     level_edu_mother == 4 ~ 'Completed high school',
                     level_edu_mother == 5 ~ 'Completed high school',
                     level_edu_mother == 6 ~ 'Completed college',
                     level_edu_mother == 7 ~ 'Completed post-graduate'))

write.csv(mat_edu, 'Data/maternal_education_20200219.csv')
```

```
##################################
### Compile data for analysis ###
##################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(tidyr)
library(lubridate)

# read in scan data (316 deconvolved scans among 50 subjects)
scan_data <- read.csv('Data/roi_percent_signal_change_20200228.csv',
                      stringsAsFactors = FALSE) %>%
  select(-c('X')) %>%
  rename('lunaid' = 'id') %>%
  mutate(date = as.Date(date, format = '%Y-%m-%d'))

# read in scored anti eye data
eye_data <- read.csv('Data/whoqol_plus_eye_20200228.csv',
                     stringsAsFactors = FALSE) %>%
  filter(whoqol_age == 'Scan') %>%
  select(-c('X', 'note', 'age_type')) %>%
  mutate(date = as.Date(date, format = '%Y-%m-%d')) %>%
  mutate(dob = as.Date(dob, format = '%Y-%m-%d'))

# read in maternal education level data
mat_edu <- read.csv('Data/maternal_education_20200219.csv',
                    stringsAsFactors = FALSE) %>%
  select(-c('X'))

# change reference category for maternal education level
mat_edu <- within(mat_edu,
                  level_edu_mother_cat <- relevel(as.factor(level_edu_mother_cat),
                                                  ref = 'Completed high school'))

# merge eye and scan data
data_for_gamms <- scan_data %>%
  left_join(eye_data, by = c('lunaid', 'date')) %>%
  mutate(sex = as.factor(sex)) %>%
  mutate(duplicate_row = duplicated(.)) %>%
  filter(duplicate_row == FALSE) %>%
  select(-c('duplicate_row')) %>%
  mutate(age_c = age - mean(age))

data_for_gamms <- data_for_gamms %>%
  left_join(mat_edu, by = c('lunaid')) %>%
  mutate(lunaid = as.factor(lunaid))

data_for_gamms %>% write.csv('Data/final_eye_scan_data_for_analysis_20200228.csv')
```

```
############
## GAMMs ##
############

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
```

```r
library(ggplot2)
library(gratia)
library(gridExtra)
library(lubridate)
library(mgcv)
library(purrr)
library(stringr)
library(tidyr)
library(xtable)


# function to plot raw data, fitted values and pointwise 95% confidence intervals
    from a GAMM
plot_gamm <- function(model, y_axis_label = '', sig = TRUE) {
  outcome <- str_split(model$gam$formula, '~')[[2]]

  newdat <- data.frame(dat$age_c) # observed centered ages
  names(newdat) <- 'age_c'

  pred <- predict.gam(model$gam,
                      newdata = newdat,
                      type = 'response',
                      se.fit = TRUE)

  plotdat <- data.frame(cbind(dat$age, pred$fit, pred$se))
  plotdat <- plotdat %>%
    mutate(outcome = 1)
  names(plotdat) <- c('age', 'fit', 'se', outcome)

  if (sig == TRUE){
    plotcolor <- '#476FD1'
  } else {
    plotcolor <- 'grey50'
  }

  gammplot <- dat %>%
    ggplot(aes(x = age,
               y = eval(parse(text = outcome)))) +
    geom_line(alpha = .2,
              color = plotcolor,
              aes(group = lunaid)) +
    geom_point(alpha = .2,
               shape = 16,
               color = plotcolor) +
    geom_ribbon(data = plotdat,
                alpha = .7,
                aes(ymin = fit-(1.96*se), ymax = fit+(1.96*se)),
                show.legend = FALSE,
                fill = plotcolor) +
    geom_line(data = plotdat,
              aes(y = fit),
              show.legend = FALSE,
              color = plotcolor) +
    labs(x = 'Age (years)',
         y = y_axis_label) +
    theme_bw() +
    theme(panel.border = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
```

```
          axis.line = element_line(colour = 'black')) +
    theme(axis.title = element_text(size = 11),
          axis.text = element_text(size = 10))
  return(gammplot)
}

# function for post-hoc analysis to identify periods of developmental change
calc_dev_change <- function(model) {
  deriv <- gratia::derivatives(model)
  deriv <- deriv %>%
    mutate(sig = !(0 > lower & 0 < upper))
  ages <- list(c(min(deriv$data[deriv$sig == TRUE]) + mean(dat$age),
                 max(deriv$data[deriv$sig == TRUE]) + mean(dat$age)))
  return(ages[[1]])
}


# read in final data file and format variables
dat <- read.csv('Data/final_eye_scan_data_for_analysis_20200228.csv',
                stringsAsFactors = FALSE)
dat <- dat %>%
  select(-c('X')) %>%
  mutate(lunaid = as.factor(lunaid)) %>%
  mutate(date = as.Date(date, format = '%Y-%m-%d')) %>%
  mutate(dob = as.Date(dob, format = '%Y-%m-%d')) %>%
  mutate(sex = as.factor(sex))
dat <- within(dat, level_edu_mother_cat <- relevel(as.factor(level_edu_mother_cat),
                                                   ref = 'Completed high school'))

### I. Behavioral outcomes

## 1. Antisaccade proportion correct trials
# fit the model with smooth term for age
model_formula <- as.formula("anti_perc_correct_nodrop ~ s(age_c, k = 10, fx = FALSE,
    bs = 'tp')")
gamm_anti_perc_correct_nodrop <- gamm(as.formula(model_formula),
                                      family = 'quasibinomial',
                                      random = list(lunaid=~age_c),
                                      data = dat,
                                      method = 'REML')
summary(gamm_anti_perc_correct_nodrop$gam)
summary(gamm_anti_perc_correct_nodrop$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_anti_perc_correct_nodrop$gam)

# identify significant periods of developmental change
anti_perc_correct_devchange <- calc_dev_change(gamm_anti_perc_correct_nodrop)
anti_perc_correct_devchange

# plot
anti_perc_correct_plot <- plot_gamm(
  model = gamm_anti_perc_correct_nodrop,
  y_axis_lab = 'Proportion correct trials',
  sig = TRUE)
anti_perc_correct_plot

# add bar to plot reflecting period of sig dev change
```

```
anti_perc_correct_plot <- anti_perc_correct_plot +
  annotate(geom = "rect",
           xmin = anti_perc_correct_devchange[1],
           xmax = anti_perc_correct_devchange[2],
           ymin = -0.04,
           ymax = -0.01,
           fill = '#476FD1')
anti_perc_correct_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("anti_perc_correct_nodrop ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_anti_perc_correct_nodrop_sex <- gamm(as.formula(model_formula),
                                          family = 'quasibinomial',
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML')
summary(gamm_anti_perc_correct_nodrop_sex$gam)
summary(gamm_anti_perc_correct_nodrop_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("anti_perc_correct_nodrop ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_anti_perc_correct_nodrop_medu <- gamm(as.formula(model_formula),
                                           family = 'quasibinomial',
                                           random = list(lunaid=~age_c),
                                           data = dat,
                                           method = 'REML')
summary(gamm_anti_perc_correct_nodrop_medu$gam)
summary(gamm_anti_perc_correct_nodrop_medu$lme)


# save random slopes and intercepts from final model
ranef_anti_perc_correct_nodrop <- ranef(gamm_anti_perc_correct_nodrop$lme)$lunaid %>%
  rename(anti_perc_correct_nodrop_rint = `(Intercept)`,
         anti_perc_correct_nodrop_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 2. Antisaccade proportion error-corrected trials
# fit the model with smooth term for age
model_formula <- as.formula("anti_perc_error_corrected_nodrop ~ s(age_c, k = 10, fx = FALSE, bs = 'tp')")
gamm_anti_perc_error_corrected_nodrop <- gamm(as.formula(model_formula),
                                              family = 'quasibinomial',
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_anti_perc_error_corrected_nodrop$gam)
summary(gamm_anti_perc_error_corrected_nodrop$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_anti_perc_error_corrected_nodrop$gam)

# identify significant periods of developmental change
```

```
anti_perc_error_corrected_nodrop_devchange <- calc_dev_change(gamm_anti_perc_error_corrected_nodrop)
anti_perc_error_corrected_nodrop_devchange

# plot
anti_perc_error_corrected_nodrop_plot <- plot_gamm(
  model = gamm_anti_perc_error_corrected_nodrop,
  y_axis_label = 'Proportion error-corrected trials',
  sig = TRUE)
anti_perc_error_corrected_nodrop_plot

anti_perc_error_corrected_nodrop_plot <- anti_perc_error_corrected_nodrop_plot +
  annotate(geom = 'rect',
           xmin = anti_perc_error_corrected_nodrop_devchange[1],
           xmax = anti_perc_error_corrected_nodrop_devchange[2],
           ymin = -0.04,
           ymax = -0.02,
           fill = '#476FD1')
anti_perc_error_corrected_nodrop_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("anti_perc_error_corrected_nodrop ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_anti_perc_error_corrected_nodrop_sex <- gamm(as.formula(model_formula),
                                                  family = 'quasibinomial',
                                                  random = list(lunaid=~age_c),
                                                  data = dat,
                                                  method = 'REML')
summary(gamm_anti_perc_error_corrected_nodrop_sex$gam)
summary(gamm_anti_perc_error_corrected_nodrop_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("anti_perc_error_corrected_nodrop ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_anti_perc_error_corrected_nodrop_medu <- gamm(as.formula(model_formula),
                                                   family = 'quasibinomial',
                                                   random = list(lunaid=~age_c),
                                                   data = dat,
                                                   method = 'REML')
summary(gamm_anti_perc_error_corrected_nodrop_medu$gam)
summary(gamm_anti_perc_error_corrected_nodrop_medu$lme)

# save random slopes and intercepts from final model
ranef_anti_perc_error_corrected_nodrop <- ranef(gamm_anti_perc_error_corrected_nodrop$lme)$lunaid %>%
  rename(anti_perc_error_corrected_nodrop_rint = `(Intercept)`,
         anti_perc_error_corrected_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 3. Average latency on correct trials
# fit the model with smooth term for age
model_formula <- as.formula("anti_avg_lat_correct_trials ~ s(age_c, k = 10, fx = FALSE, bs = 'tp')")
gamm_anti_avg_lat_correct_trials <- gamm(as.formula(model_formula),
                                         random = list(lunaid=~age_c),
                                         data = dat,
                                         method = 'REML')
```

```r
summary(gamm_anti_avg_lat_correct_trials$gam)
summary(gamm_anti_avg_lat_correct_trials$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_anti_avg_lat_correct_trials$gam)

# identify significant periods of developmental change
anti_avg_lat_correct_trials_devchange <- calc_dev_change(gamm_anti_avg_lat_correct_trials)
anti_avg_lat_correct_trials_devchange

# plot
anti_avg_lat_correct_trials_plot <- plot_gamm(
  model = gamm_anti_avg_lat_correct_trials,
  y_axis_label = 'Average latency correct trials',
  sig = TRUE)
anti_avg_lat_correct_trials_plot

anti_avg_lat_correct_trials_plot <- anti_avg_lat_correct_trials_plot +
  annotate(geom = 'rect',
           xmin = anti_avg_lat_correct_trials_devchange[1],
           xmax = anti_avg_lat_correct_trials_devchange[2],
           ymin = 300,
           ymax = 310,
           fill = '#476FD1')
anti_avg_lat_correct_trials_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("anti_avg_lat_correct_trials ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_anti_avg_lat_correct_trials_sex <- gamm(as.formula(model_formula),
                                             random = list(lunaid=~age_c),
                                             data = dat,
                                             method = 'REML')
summary(gamm_anti_avg_lat_correct_trials_sex$gam)
summary(gamm_anti_avg_lat_correct_trials_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("anti_avg_lat_correct_trials ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_anti_avg_lat_correct_trials_medu <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_anti_avg_lat_correct_trials_medu$gam)
summary(gamm_anti_avg_lat_correct_trials_medu$lme)


# save random slopes and intercepts from final model
ranef_anti_avg_lat_correct_trials <- ranef(gamm_anti_avg_lat_correct_trials$lme)$lunaid %>%
  rename(anti_avg_lat_correct_trials_rint = `(Intercept)`,
         anti_avg_lat_correct_trials_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 4. Average latency on error-corrected trials
```

```
# fit the model with smooth term for age
model_formula <- as.formula("anti_avg_lat_error_corrected_trials ~ s(age_c, k = 10,
    fx = FALSE, bs = 'tp')")
gamm_anti_avg_lat_error_corrected_trials <- gamm(model_formula,
                                                 random = list(lunaid=~age_c),
                                                 data = dat,
                                                 method = 'REML')
summary(gamm_anti_avg_lat_error_corrected_trials$gam)
summary(gamm_anti_avg_lat_error_corrected_trials$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_anti_avg_lat_error_corrected_trials$gam)

# plot
anti_avg_lat_error_corrected_trials_plot <- plot_gamm(
  model = gamm_anti_avg_lat_error_corrected_trials,
  y_axis_label = 'Average latency error-corrected trials',
  sig = FALSE)
anti_avg_lat_error_corrected_trials_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("anti_avg_lat_error_corrected_trials ~ s(age_c, k = 10,
    fx = FALSE, bs = 'tp') + sex")
gamm_anti_avg_lat_error_corrected_trials_sex <- gamm(as.formula(model_formula),
                                                     random = list(lunaid=~age_c),
                                                     data = dat,
                                                     method = 'REML')
summary(gamm_anti_avg_lat_error_corrected_trials_sex$gam)
summary(gamm_anti_avg_lat_error_corrected_trials_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("anti_avg_lat_error_corrected_trials ~ s(age_c, k = 10,
    fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_anti_avg_lat_error_corrected_trials_medu <- gamm(as.formula(model_formula),
                                                      random = list(lunaid=~age_c),
                                                      data = dat,
                                                      method = 'REML')
summary(gamm_anti_avg_lat_error_corrected_trials_medu$gam)
summary(gamm_anti_avg_lat_error_corrected_trials_medu$lme)


# save random slopes and intercepts from final model
ranef_anti_avg_lat_error_corrected_trials <- ranef(gamm_anti_avg_lat_error_corrected
    _trials$lme)$lunaid %>%
  rename(anti_avg_lat_error_corrected_trials_rint = `(Intercept)`,
         anti_avg_lat_error_corrected_trials_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


### II. Brain function outcomes - antisaccade correct trials vs. baseline
## 1. SEF
# fit the model with smooth term for age
model_formula <- as.formula("SEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE,
     bs = 'tp')")
gamm_SEF_anti_corr_v_baseline <- gamm(model_formula,
                                      random = list(lunaid=~age_c),
```

```
                                          data = dat,
                                          method = 'REML')
summary(gamm_SEF_anti_corr_v_baseline$gam)
summary(gamm_SEF_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_SEF_anti_corr_v_baseline$gam)

# plot
SEF_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_SEF_anti_corr_v_baseline,
  y_axis_label = 'SEF percent signal change',
  sig = FALSE)
SEF_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("SEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE,
    bs = 'tp') + sex")
gamm_SEF_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML')
summary(gamm_SEF_anti_corr_v_baseline_sex$gam)
summary(gamm_SEF_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("SEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE,
    bs = 'tp') + level_edu_mother_cat")
gamm_SEF_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                           random = list(lunaid=~age_c),
                                           data = dat,
                                           method = 'REML')
summary(gamm_SEF_anti_corr_v_baseline_medu$gam)
summary(gamm_SEF_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts
ranef_SEF_anti_corr_v_baseline <- ranef(gamm_SEF_anti_corr_v_baseline$lme)$lunaid
  %>%
  rename(SEF_anti_corr_v_baseline_rint = `(Intercept)`,
         SEF_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 2. pre-SMA
# fit the model with smooth term for age
model_formula <- as.formula("pre_SMA_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_pre_SMA_anti_corr_v_baseline <- gamm(model_formula,
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML')
summary(gamm_pre_SMA_anti_corr_v_baseline$gam)
summary(gamm_pre_SMA_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
```

```
gam.check(gamm_pre_SMA_anti_corr_v_baseline$gam)

# plot
pre_SMA_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_pre_SMA_anti_corr_v_baseline,
  y_axis_label = 'Pre-SMA percent signal change',
  sig = FALSE)
pre_SMA_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("pre_SMA_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_pre_SMA_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_pre_SMA_anti_corr_v_baseline_sex$gam)
summary(gamm_pre_SMA_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("pre_SMA_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_pre_SMA_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                               random = list(lunaid=~age_c),
                                               data = dat,
                                               method = 'REML')
summary(gamm_pre_SMA_anti_corr_v_baseline_medu$gam)
summary(gamm_pre_SMA_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_pre_SMA_anti_corr_v_baseline <- ranef(gamm_pre_SMA_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(pre_SMA_anti_corr_v_baseline_rint = `(Intercept)`,
         pre_SMA_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 3. L FEF
# fit the model with smooth term for age
model_formula <- as.formula("L_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_L_FEF_anti_corr_v_baseline <- gamm(model_formula,
                                        random = list(lunaid=~age_c),
                                        data = dat,
                                        method = 'REML')
summary(gamm_L_FEF_anti_corr_v_baseline$gam)
summary(gamm_L_FEF_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_L_FEF_anti_corr_v_baseline$gam)

# identify significant periods of developmental change
L_FEF_devchange <- calc_dev_change(gamm_L_FEF_anti_corr_v_baseline)
L_FEF_devchange

# plot
```

```
L_FEF_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_L_FEF_anti_corr_v_baseline,
  y_axis_label = 'L FEF percent signal change')
L_FEF_anti_corr_v_baseline_plot

L_FEF_anti_corr_v_baseline_plot <- L_FEF_anti_corr_v_baseline_plot +
  annotate(geom = 'rect',
           xmin = L_FEF_devchange[1],
           xmax = L_FEF_devchange[2],
           ymin =  -5, ymax = -4.5,
           fill = '#476FD1')
L_FEF_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("L_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_L_FEF_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML')
summary(gamm_L_FEF_anti_corr_v_baseline_sex$gam)
summary(gamm_L_FEF_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("L_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_L_FEF_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                             random = list(lunaid=~age_c),
                                             data = dat,
                                             method = 'REML')
summary(gamm_L_FEF_anti_corr_v_baseline_medu$gam)
summary(gamm_L_FEF_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_L_FEF_anti_corr_v_baseline <- ranef(gamm_L_FEF_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(L_FEF_anti_corr_v_baseline_rint = `(Intercept)`,
         L_FEF_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 4. R FEF
# fit the model with smooth term for age
model_formula <- as.formula("R_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_R_FEF_anti_corr_v_baseline <- gamm(model_formula,
                                        random = list(lunaid=~age_c),
                                        data = dat,
                                        method = 'REML',
                                        control = lmeControl(maxIter = 50))
summary(gamm_R_FEF_anti_corr_v_baseline$gam)
summary(gamm_R_FEF_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_R_FEF_anti_corr_v_baseline$gam)

```

```r
# plot
R_FEF_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_R_FEF_anti_corr_v_baseline,
  y_axis_label = 'R FEF percent signal change',
  sig = FALSE)
R_FEF_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("R_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_R_FEF_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML',
                                            control = lmeControl(opt = 'optim'))
summary(gamm_R_FEF_anti_corr_v_baseline_sex$gam)
summary(gamm_R_FEF_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("R_FEF_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_R_FEF_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                             random = list(lunaid=~age_c),
                                             data = dat,
                                             method = 'REML',
                                             control = lmeControl(opt = 'optim'))
summary(gamm_R_FEF_anti_corr_v_baseline_medu$gam)
summary(gamm_R_FEF_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts
ranef_R_FEF_anti_corr_v_baseline <- ranef(gamm_R_FEF_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(R_FEF_anti_corr_v_baseline_rint = `(Intercept)`,
         R_FEF_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 5. L putamen
# fit the model with smooth term for age
model_formula <- as.formula("L_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_L_putamen_anti_corr_v_baseline <- gamm(model_formula,
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML',
                                            control = lmeControl(maxIter = 50))
summary(gamm_L_putamen_anti_corr_v_baseline$gam)
summary(gamm_L_putamen_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_L_putamen_anti_corr_v_baseline$gam)

# plot
L_putamen_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_L_putamen_anti_corr_v_baseline,
  y_axis_label = 'L putamen percent signal change',
  sig = FALSE)
```

```r
L_putamen_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("L_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_L_putamen_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                                random = list(lunaid=~age_c),
                                                data = dat,
                                                method = 'REML')
summary(gamm_L_putamen_anti_corr_v_baseline_sex$gam)
summary(gamm_L_putamen_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("L_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_L_putamen_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                                 random = list(lunaid=~age_c),
                                                 data = dat,
                                                 method = 'REML')
summary(gamm_L_putamen_anti_corr_v_baseline_medu$gam)
summary(gamm_L_putamen_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts
ranef_L_putamen_anti_corr_v_baseline <- ranef(gamm_L_putamen_anti_corr_v_baseline$
    lme)$lunaid %>%
  rename(L_putamen_anti_corr_v_baseline_rint = `(Intercept)`,
         L_putamen_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 6. R putamen
# fit the model with smooth term for age
model_formula <- as.formula("R_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_R_putamen_anti_corr_v_baseline <- gamm(model_formula,
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML',
                                            control = lmeControl(maxIter = 50))
summary(gamm_R_putamen_anti_corr_v_baseline$gam)
summary(gamm_R_putamen_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_R_putamen_anti_corr_v_baseline$gam)

# plot
R_putamen_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_R_putamen_anti_corr_v_baseline,
  y_axis_label = 'R putamen percent signal change',
  sig = FALSE
)
R_putamen_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("R_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
```

```
    FALSE, bs = 'tp') + sex")
gamm_R_putamen_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                                random = list(lunaid=~age_c),
                                                data = dat,
                                                method = 'REML')
summary(gamm_R_putamen_anti_corr_v_baseline_sex$gam)
summary(gamm_R_putamen_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("R_putamen_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_R_putamen_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                                 random = list(lunaid=~age_c),
                                                 data = dat,
                                                 method = 'REML')
summary(gamm_R_putamen_anti_corr_v_baseline_medu$gam)
summary(gamm_R_putamen_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_R_putamen_anti_corr_v_baseline <- ranef(gamm_R_putamen_anti_corr_v_baseline$
    lme)$lunaid %>%
  rename(R_putamen_anti_corr_v_baseline_rint = `(Intercept)`,
         R_putamen_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 7. L pPC
# fit the model with smooth term for age
model_formula <- as.formula("L_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_L_pPC_anti_corr_v_baseline <- gamm(model_formula,
                                        random = list(lunaid=~age_c),
                                        data = dat,
                                        method = 'REML',
                                        control = lmeControl(maxIter = 50))
summary(gamm_L_pPC_anti_corr_v_baseline$gam)
summary(gamm_L_pPC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_L_pPC_anti_corr_v_baseline$gam)

# identify significant periods of developmental change
L_pPC_devchange <- calc_dev_change(gamm_L_pPC_anti_corr_v_baseline)
L_pPC_devchange

# plot
L_pPC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_L_pPC_anti_corr_v_baseline,
  y_axis_label = 'L pPC percent signal change',
  sig = TRUE)
L_pPC_anti_corr_v_baseline_plot

L_pPC_anti_corr_v_baseline_plot <- L_pPC_anti_corr_v_baseline_plot +
  annotate(geom = 'rect',
           xmin = L_pPC_devchange[1],
           xmax = L_pPC_devchange[2],
           ymin = -8,
```

```
            ymax = -7.5,
            fill = '#476FD1')
L_pPC_anti_corr_v_baseline_plot


# fit the model w/ fixed effects of sex
model_formula <- as.formula("L_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_L_pPC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML')
summary(gamm_L_pPC_anti_corr_v_baseline_sex$gam)
summary(gamm_L_pPC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("L_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_L_pPC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                             random = list(lunaid=~age_c),
                                             data = dat,
                                             method = 'REML')
summary(gamm_L_pPC_anti_corr_v_baseline_medu$gam)
summary(gamm_L_pPC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_L_pPC_anti_corr_v_baseline <- ranef(gamm_L_pPC_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(L_pPC_anti_corr_v_baseline_rint = `(Intercept)`,
         L_pPC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 8. R pPC
# fit the model with smooth term for age
model_formula <- as.formula("R_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_R_pPC_anti_corr_v_baseline <- gamm(model_formula,
                                        random = list(lunaid=~age_c),
                                        data = dat,
                                        method = 'REML',
                                        control = lmeControl(maxIter = 50))
summary(gamm_R_pPC_anti_corr_v_baseline$gam)
summary(gamm_R_pPC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_R_pPC_anti_corr_v_baseline$gam)

# identify significant periods of developmental change
R_pPC_devchange <- calc_dev_change(gamm_R_pPC_anti_corr_v_baseline)
R_pPC_devchange

# plot
R_pPC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_R_pPC_anti_corr_v_baseline,
  y_axis_label = 'R pPC percent signal change',
  sig = TRUE)
```

```
R_pPC_anti_corr_v_baseline_plot

R_pPC_anti_corr_v_baseline_plot <- R_pPC_anti_corr_v_baseline_plot +
  annotate(geom = 'rect',
           xmin = R_pPC_devchange[1],
           xmax = R_pPC_devchange[2],
           ymin = -8.5,
           ymax = -8,
           fill = '#476FD1')
R_pPC_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("R_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_R_pPC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML')
summary(gamm_R_pPC_anti_corr_v_baseline_sex$gam)
summary(gamm_R_pPC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("R_pPC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_R_pPC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                             random = list(lunaid=~age_c),
                                             data = dat,
                                             method = 'REML')
summary(gamm_R_pPC_anti_corr_v_baseline_medu$gam)
summary(gamm_R_pPC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts
ranef_R_pPC_anti_corr_v_baseline <- ranef(gamm_R_pPC_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(R_pPC_anti_corr_v_baseline_rint = `(Intercept)`,
         R_pPC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 9. L dlPFC
# fit the model with smooth term for age
model_formula <- as.formula("L_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_L_dlPFC_anti_corr_v_baseline <- gamm(model_formula,
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML',
                                          control = lmeControl(maxIter = 50))
summary(gamm_L_dlPFC_anti_corr_v_baseline$gam)
summary(gamm_L_dlPFC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_L_dlPFC_anti_corr_v_baseline$gam)

# plot
L_dlPFC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_L_dlPFC_anti_corr_v_baseline,
```

```
  y_axis_label = 'L dlPFC percent signal change',
  sig = FALSE)
L_dlPFC_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("L_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_L_dlPFC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_L_dlPFC_anti_corr_v_baseline_sex$gam)
summary(gamm_L_dlPFC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("L_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_L_dlPFC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                               random = list(lunaid=~age_c),
                                               data = dat,
                                               method = 'REML')
summary(gamm_L_dlPFC_anti_corr_v_baseline_medu$gam)
summary(gamm_L_dlPFC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts
ranef_L_dlPFC_anti_corr_v_baseline <- ranef(gamm_L_dlPFC_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(L_dlPFC_anti_corr_v_baseline_rint = `(Intercept)`,
         L_dlPFC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 10. R dlPFC
# fit the model with smooth term for age
model_formula <- as.formula("R_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_R_dlPFC_anti_corr_v_baseline <- gamm(model_formula,
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML',
                                          control = lmeControl(maxIter = 50))
summary(gamm_R_dlPFC_anti_corr_v_baseline$gam)
summary(gamm_R_dlPFC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_R_dlPFC_anti_corr_v_baseline$gam)

# identify significant periods of developmental change
R_dlPFC_devchange <- calc_dev_change(gamm_R_dlPFC_anti_corr_v_baseline)
R_dlPFC_devchange

# plot
R_dlPFC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_R_dlPFC_anti_corr_v_baseline,
  y_axis_label = 'R dlPFC percent signal change',
  sig = TRUE)
R_dlPFC_anti_corr_v_baseline_plot
```

```
837
838 R_dlPFC_anti_corr_v_baseline_plot <- R_dlPFC_anti_corr_v_baseline_plot +
839   annotate(geom = 'rect',
840            xmin = R_dlPFC_devchange[1],
841            xmax = R_dlPFC_devchange[2],
842            ymin = -8.5, ymax = -8,
843            fill = '#476FD1')
844 R_dlPFC_anti_corr_v_baseline_plot
845
846 # fit the model w/ fixed effects of sex
847 model_formula <- as.formula("R_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
      FALSE, bs = 'tp') + sex")
848 gamm_R_dlPFC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
849                                               random = list(lunaid=~age_c),
850                                               data = dat,
851                                               method = 'REML')
852 summary(gamm_R_dlPFC_anti_corr_v_baseline_sex$gam)
853 summary(gamm_R_dlPFC_anti_corr_v_baseline_sex$lme)
854
855 # fit the model w/ fixed effects of maternal education level
856 model_formula <- as.formula("R_dlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
      FALSE, bs = 'tp') + level_edu_mother_cat")
857 gamm_R_dlPFC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
858                                                random = list(lunaid=~age_c),
859                                                data = dat,
860                                                method = 'REML')
861 summary(gamm_R_dlPFC_anti_corr_v_baseline_medu$gam)
862 summary(gamm_R_dlPFC_anti_corr_v_baseline_medu$lme)
863
864 # save random slopes and intercepts from final model
865 ranef_R_dlPFC_anti_corr_v_baseline <- ranef(gamm_R_dlPFC_anti_corr_v_baseline$lme)$
      lunaid %>%
866   rename(R_dlPFC_anti_corr_v_baseline_rint = `(Intercept)`,
867          R_dlPFC_anti_corr_v_baseline_rslope = age_c) %>%
868   tibble::rownames_to_column(., 'lunaid') %>%
869   mutate(lunaid = as.factor(gsub('1/', '', lunaid)))
870
871
872 ## 11. L vlPFC
873 # fit the model with smooth term for age
874 model_formula <- as.formula("L_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
      FALSE, bs = 'tp')")
875 gamm_L_vlPFC_anti_corr_v_baseline <- gamm(model_formula,
876                                           random = list(lunaid=~age_c),
877                                           data = dat,
878                                           method = 'REML',
879                                           control = lmeControl(maxIter = 100,
                                                msMaxIter = 100, niterEM = 50, opt = '
                                                optim'))
880 summary(gamm_L_vlPFC_anti_corr_v_baseline$gam)
881 summary(gamm_L_vlPFC_anti_corr_v_baseline$lme)
882
883 # model diagnostics
884 par(mfrow = c(2,2))
885 gam.check(gamm_L_vlPFC_anti_corr_v_baseline$gam)
886
887 # plot
888 L_vlPFC_anti_corr_v_baseline_plot <- plot_gamm(
889   model = gamm_L_vlPFC_anti_corr_v_baseline,
```

```
  y_axis_label = 'L vlPFC percent signal change',
  sig = FALSE)
L_vlPFC_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("L_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + sex")
gamm_L_vlPFC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML',
                                              control = lmeControl(opt = 'optim'))
summary(gamm_L_vlPFC_anti_corr_v_baseline_sex$gam)
summary(gamm_L_vlPFC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("L_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_L_vlPFC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                               random = list(lunaid=~age_c),
                                               data = dat,
                                               method = 'REML',
                                               control = lmeControl(opt = 'optim'))
summary(gamm_L_vlPFC_anti_corr_v_baseline_medu$gam)
summary(gamm_L_vlPFC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_L_vlPFC_anti_corr_v_baseline <- ranef(gamm_L_vlPFC_anti_corr_v_baseline$lme)$
    lunaid %>%
  rename(L_vlPFC_anti_corr_v_baseline_rint = `(Intercept)`,
         L_vlPFC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 12. R vlPFC
# fit the model with smooth term for age
model_formula <- as.formula("R_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp')")
gamm_R_vlPFC_anti_corr_v_baseline <- gamm(model_formula,
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML',
                                          control = lmeControl(maxIter = 50))
summary(gamm_R_vlPFC_anti_corr_v_baseline$gam)
summary(gamm_R_vlPFC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_R_vlPFC_anti_corr_v_baseline$gam)

# plot
R_vlPFC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_R_vlPFC_anti_corr_v_baseline,
  y_axis_label = 'R vlPFC percent signal change',
  sig = FALSE)
R_vlPFC_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of sex
```

```r
model_formula <- as.formula("R_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_R_vlPFC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_R_vlPFC_anti_corr_v_baseline_sex$gam)
summary(gamm_R_vlPFC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("R_vlPFC_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_R_vlPFC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                               random = list(lunaid=~age_c),
                                               data = dat,
                                               method = 'REML')
summary(gamm_R_vlPFC_anti_corr_v_baseline_medu$gam)
summary(gamm_R_vlPFC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts from final model
ranef_R_vlPFC_anti_corr_v_baseline <- ranef(gamm_R_vlPFC_anti_corr_v_baseline$lme)$lunaid %>%
  rename(R_vlPFC_anti_corr_v_baseline_rint = `(Intercept)`,
         R_vlPFC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## 13. dACC
# fit the model with smooth term for age
model_formula <- as.formula("dACC_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp')")
gamm_dACC_anti_corr_v_baseline <- gamm(model_formula,
                                       random = list(lunaid=~age_c),
                                       data = dat,
                                       method = 'REML',
                                       control = lmeControl(maxIter = 50))
summary(gamm_dACC_anti_corr_v_baseline$gam)
summary(gamm_dACC_anti_corr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_dACC_anti_corr_v_baseline$gam)

# plot
dACC_anti_corr_v_baseline_plot <- plot_gamm(
  model = gamm_dACC_anti_corr_v_baseline,
  y_axis_label = 'dACC percent signal change correct trials',
  sig = FALSE)
dACC_anti_corr_v_baseline_plot

# fit the model w/ fixed effects of gender
model_formula <- as.formula("dACC_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_dACC_anti_corr_v_baseline_sex <- gamm(as.formula(model_formula),
                                           random = list(lunaid=~age_c),
                                           data = dat,
                                           method = 'REML')
summary(gamm_dACC_anti_corr_v_baseline_sex$gam)
```

```r
summary(gamm_dACC_anti_corr_v_baseline_sex$lme)

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("dACC_anti_corr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_dACC_anti_corr_v_baseline_medu <- gamm(as.formula(model_formula),
                                            random = list(lunaid=~age_c),
                                            data = dat,
                                            method = 'REML')
summary(gamm_dACC_anti_corr_v_baseline_medu$gam)
summary(gamm_dACC_anti_corr_v_baseline_medu$lme)

# save random slopes and intercepts, adjusting for maternal education
ranef_dACC_anti_corr_v_baseline <- ranef(gamm_dACC_anti_corr_v_baseline_medu$lme)$lunaid %>%
  rename(dACC_anti_corr_v_baseline_rint = `(Intercept)`,
         dACC_anti_corr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


### III. Brain function outcomes - antisaccade error-corrected trials vs. baseline
## 1. dACC
# fit the model with smooth term for age
model_formula <- as.formula("dACC_anti_errcorr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp')")
gamm_dACC_anti_errcorr_v_baseline <- gamm(model_formula,
                                          random = list(lunaid=~age_c),
                                          data = dat,
                                          method = 'REML',
                                          control = lmeControl(opt = 'optim'))
summary(gamm_dACC_anti_errcorr_v_baseline$gam)
summary(gamm_dACC_anti_errcorr_v_baseline$lme)

# model diagnostics
par(mfrow = c(2,2))
gam.check(gamm_dACC_anti_errcorr_v_baseline$gam)

# identify significant periods of developmental change
dACC_errcorr_devchange <- calc_dev_change(gamm_dACC_anti_errcorr_v_baseline)
dACC_errcorr_devchange

# plot
dACC_anti_errcorr_v_baseline_plot <- plot_gamm(
  model = gamm_dACC_anti_errcorr_v_baseline,
  y_axis_label = 'dACC percent signal change error-corrected trials',
  sig = TRUE)
dACC_anti_errcorr_v_baseline_plot

# fit the model w/ fixed effects of sex
model_formula <- as.formula("dACC_anti_errcorr_v_baseline ~ s(age_c, k = 10, fx = FALSE, bs = 'tp') + sex")
gamm_dACC_anti_errcorr_v_baseline_sex <- gamm(as.formula(model_formula),
                                              random = list(lunaid=~age_c),
                                              data = dat,
                                              method = 'REML')
summary(gamm_dACC_anti_errcorr_v_baseline_sex$gam)
summary(gamm_dACC_anti_errcorr_v_baseline_sex$lme)

```

```r
# identify significant periods of developmental change, adjusting for sex
dACC_errcorr_sex_devchange <- calc_dev_change(gamm_dACC_anti_errcorr_v_baseline_sex)
dACC_errcorr_sex_devchange

dACC_anti_errcorr_v_baseline_plot <- dACC_anti_errcorr_v_baseline_plot +
  annotate(geom = 'rect',
           xmin = dACC_errcorr_sex_devchange[1],
           xmax = dACC_errcorr_sex_devchange[2],
           ymin =  -25,
           ymax = -24,
           fill = '#476FD1')
dACC_anti_errcorr_v_baseline_plot

# fit the model w/ fixed effects of maternal education level
model_formula <- as.formula("dACC_anti_errcorr_v_baseline ~ s(age_c, k = 10, fx =
    FALSE, bs = 'tp') + level_edu_mother_cat")
gamm_dACC_anti_errcorr_v_baseline_medu <- gamm(as.formula(model_formula),
                                               random = list(lunaid=~age_c),
                                               data = dat,
                                               method = 'REML')
summary(gamm_dACC_anti_errcorr_v_baseline_medu$gam)
summary(gamm_dACC_anti_errcorr_v_baseline_medu$lme)

# save random slopes and intercepts, adjusting for sex
ranef_dACC_anti_errcorr_v_baseline <- ranef(gamm_dACC_anti_errcorr_v_baseline_sex$
    lme)$lunaid %>%
  rename(dACC_anti_errcorr_v_baseline_rint = `(Intercept)`,
         dACC_anti_errcorr_v_baseline_rslope = age_c) %>%
  tibble::rownames_to_column(., 'lunaid') %>%
  mutate(lunaid = as.factor(gsub('1/', '', lunaid)))


## test of global effect of maternal education level for each model
mods <- c('gamm_anti_perc_correct_nodrop',
          'gamm_anti_perc_error_corrected_nodrop',
          'gamm_anti_avg_lat_correct_trials',
          "gamm_anti_avg_lat_error_corrected_trials",
          "gamm_SEF_anti_corr_v_baseline",
          "gamm_pre_SMA_anti_corr_v_baseline",
          "gamm_L_FEF_anti_corr_v_baseline",
          "gamm_R_FEF_anti_corr_v_baseline",
          "gamm_L_putamen_anti_corr_v_baseline",
          "gamm_R_putamen_anti_corr_v_baseline",
          "gamm_L_pPC_anti_corr_v_baseline",
          "gamm_R_pPC_anti_corr_v_baseline",
          "gamm_L_dlPFC_anti_corr_v_baseline",
          "gamm_R_dlPFC_anti_corr_v_baseline",
          "gamm_L_vlPFC_anti_corr_v_baseline",
          "gamm_R_vlPFC_anti_corr_v_baseline",
          "gamm_dACC_anti_corr_v_baseline",
          "gamm_dACC_anti_errcorr_v_baseline")

global_test_medu <- matrix(nrow = length(mods), ncol = 4)
for(i in 1:length(mods)){
  mod_medu <- paste0(mods[i], '_medu')
  global_test_medu[i, 1] <- mod_medu
  global_test_medu[i, 2:4] <- anova(get(mod_medu)$gam)$pTerms.table[1:3]
}
colnames(global_test_medu) <- c('model', 'df', 'F', 'p-value for global medu')
```

```
global_test_medu
print(xtable(global_test_medu, type = "latex"),
      file = "gamm_test_global_sig_medu_terms.tex")

## tests of effect of sex for each model
test_sex <- matrix(nrow = length(mods), ncol = 4)
for (i in 1:length(mods)){
  mod_sex <- paste0(mods[i], '_sex')
  test_sex[i, 1] <- mod_sex
  test_sex[i, 2:4] <- summary(get(mod_sex)$gam)$p.table[2, c(1, 3:4)]
}
colnames(test_sex) <- c('model', 'estimate', 't', 'p-value for sex')
test_sex
print(xtable(test_sex, type = "latex"),
      file = "gamm_test_sig_sex_terms.tex")


### IV. Compile results and plots
# summary of significance of smooth terms for age for final models
mods <- c('gamm_anti_perc_correct_nodrop',
          'gamm_anti_perc_error_corrected_nodrop',
          'gamm_anti_avg_lat_correct_trials',
          "gamm_anti_avg_lat_error_corrected_trials",
          "gamm_SEF_anti_corr_v_baseline",
          "gamm_pre_SMA_anti_corr_v_baseline",
          "gamm_L_FEF_anti_corr_v_baseline",
          "gamm_R_FEF_anti_corr_v_baseline",
          "gamm_L_putamen_anti_corr_v_baseline",
          "gamm_R_putamen_anti_corr_v_baseline",
          "gamm_L_pPC_anti_corr_v_baseline",
          "gamm_R_pPC_anti_corr_v_baseline",
          "gamm_L_dlPFC_anti_corr_v_baseline",
          "gamm_R_dlPFC_anti_corr_v_baseline",
          "gamm_L_vlPFC_anti_corr_v_baseline",
          "gamm_R_vlPFC_anti_corr_v_baseline",
          "gamm_dACC_anti_corr_v_baseline",
          "gamm_dACC_anti_errcorr_v_baseline_sex")
sumtab <- sapply(mods, function (x) summary(get(x)$gam)$s.table)
rownames(sumtab) <- c('edf', 'Ref.df', 'F', 'p-value')
sumtab <- as.data.frame(t(sumtab)) %>%
  tibble::rownames_to_column(., 'model') %>%
  select(-c('Ref.df'))
sumtab <- cbind(sumtab, p.adjust(sumtab$`p-value`, method = 'fdr'))
names(sumtab)[ncol(sumtab)] <- 'q-value'
sumtab <- sumtab %>%
  mutate(survive_multcomp = case_when(`q-value` < 0.05 ~ 1,
                                      `q-value` >= 0.05 ~ 0))
print(xtable(sumtab %>% select(-c(survive_multcomp)),
             digits = c(0,0,2,2,7,7),
             type = "latex"),
      file = "sumtab_age_terms.tex")

# merge random effects and save for further analysis
random_effects_list <- do.call('list', mget(grep('ranef', names(.GlobalEnv), value =
    TRUE)))
random_effects <- random_effects_list %>%
  purrr::reduce(left_join, by = 'lunaid')
write.csv(random_effects, 'Data/random_effects_from_gamms_20200312.csv')

```

```
## plots
# behavioral (accuracy and latency) plot
behav_plot <- grid.arrange(
  anti_perc_correct_plot,
  anti_avg_lat_correct_trials_plot,
  anti_perc_error_corrected_nodrop_plot,
  anti_avg_lat_error_corrected_trials_plot,
  ncol = 2)
behav_plot
ggsave(filename = 'Plots/behav_plot.png',
       plot = behav_plot,
       width = 7, height = 7, units = 'in')

# motor response ROIs
motor_resp_rois_plot <- grid.arrange(
  SEF_anti_corr_v_baseline_plot,
  pre_SMA_anti_corr_v_baseline_plot,
  L_FEF_anti_corr_v_baseline_plot,
  R_FEF_anti_corr_v_baseline_plot,
  L_pPC_anti_corr_v_baseline_plot,
  R_pPC_anti_corr_v_baseline_plot,
  L_putamen_anti_corr_v_baseline_plot,
  R_putamen_anti_corr_v_baseline_plot,
  ncol = 2)
motor_resp_rois_plot
ggsave(filename = 'Plots/motor_rois.png',
       plot = motor_resp_rois_plot,
       width = 8, height = 10.5, units = 'in')

# executive control rois
exec_rois_plot <- grid.arrange(
  L_vlPFC_anti_corr_v_baseline_plot,
  R_vlPFC_anti_corr_v_baseline_plot,
  L_dlPFC_anti_corr_v_baseline_plot,
  R_dlPFC_anti_corr_v_baseline_plot,
  ncol = 2)
exec_rois_plot
ggsave(filename = 'Plots/exec_rois.png',
       plot = exec_rois_plot,
       width = 7, height = 6, units = 'in')

# dACC roi
dACC_plot <- grid.arrange(
  dACC_anti_corr_v_baseline_plot,
  dACC_anti_errcorr_v_baseline_plot,
  ncol = 2)
dACC_plot
ggsave(filename = 'Plots/dACC.png',
       plot = dACC_plot,
       width = 8, height = 4, units = 'in')
```

```
########################################
### Bootstrap-enhanced elastic net ###
########################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(caret)
```

```r
library(dplyr)
library(ensr)
library(ggplot2)
library(glmnet)
library(gridExtra)
library(lubridate)
library(purrr)
library(tidyr)
library(xtable)

# response measure (whoqol score)
whoqol <- read.csv('Data/whoqol_ages_scores_20200204.csv',
                   stringsAsFactors = FALSE) %>%
  mutate(date = as.Date(date)) %>%
  select(-c('X'))

# predictors (random effects from GAMM models)
ranefs <- read.csv('Data/random_effects_from_gamms_20200312.csv',
                   stringsAsFactors = FALSE) %>%
  select(-c(X))

# scale predictors and response variable
ranefs <- ranefs %>%
  mutate_at(vars(matches('anti')), scale)
head(ranefs)

# join response and predictors
dat <- whoqol %>%
  right_join(ranefs, by = 'lunaid') %>%
  # z-score WHO-QOL domain scores
  mutate(D_1_Raw_z = scale(D_1_Raw)) %>%
  mutate(D_2_Raw_z = scale(D_2_Raw)) %>%
  mutate(D_3_Raw_z = scale(D_3_Raw)) %>%
  mutate(D_4_Raw_z = scale(D_4_Raw)) %>%
  # for each subject, average their four z-scored domain scores
  rowwise() %>%
  mutate(whoqol_avg_score = mean(c(D_1_Raw_z, D_2_Raw_z, D_3_Raw_z, D_4_Raw_z)))

whoqol_avg_score_z <- scale(dat$whoqol_avg_score)[1:50]

dat <- cbind(dat, whoqol_avg_score_z) %>%
  select(-(date:counter)) %>%
  select(-(D_1_Raw_z:whoqol_avg_score))

write.csv(dat, 'Data/data_for_elastic_net_20200317.csv')

# function to plot histogram of whoqol scores
plot_whoqol <- function(x_var, x_axis_label, binw, centering) {
  p <- ggplot(dat2, aes_string(x = x_var)) +
    geom_histogram(color = 'black',
                   fill = 'white',
                   binwidth = binw,
                   center = centering) +
    labs(x = x_axis_label,
         y = 'Frequency') +
    theme_bw() +
    theme(panel.border = element_blank(), panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(), axis.line = element_line(colour = '
              black')) +
```

```
    theme(axis.title = element_text(size = 14),
          axis.text = element_text(size = 11))
  return(p)
}

# histograms of individual domain scores and composite scores
dom1 <- plot_whoqol(x_var = 'D_1_Raw',
                    x_axis_label = 'Physical health',
                    binw = 2, centering = 1)
dom2 <- plot_whoqol(x_var = 'D_2_Raw',
                    x_axis_label = 'Psychological health',
                    binw = 2, centering = 0)
dom3 <- plot_whoqol(x_var = 'D_3_Raw',
                    x_axis_label = 'Social',
                    binw = 1, centering = 1) +
  scale_y_continuous(limits = c(0, 13))
dom4 <- plot_whoqol(x_var = 'D_4_Raw',
                    x_axis_label = 'Environment',
                    binw = 2, centering = 0) +
  scale_y_continuous(limits = c(0, 11))
composite <- plot_whoqol(x_var = 'whoqol_avg_score_z',
                         x_axis_label = 'Standardized composite score',
                         binw = 0.5, centering = 0.75) +
  scale_y_continuous(limits = c(0, 13))

# arrange histograms in single plot
layoutmatrix <- rbind(c(1,2,3),
                      c(1,2,4),
                      c(5,6,4),
                      c(5,6,7))
whoqol_hists <- gridExtra::grid.arrange(dom1, dom2, grid::nullGrob(),
                                        composite,
                                        dom3, dom4, grid::nullGrob(),
                                        layout_matrix = layoutmatrix,
                                        ncol = 3)
whoqol_hists
ggsave('whoqol_hists.png', whoqol_hists, width = 10, height = 7, units = 'in')


## prep data for elastic net
# split data into response y and predictors X
set.seed(100)
X <- as.matrix(dat %>% select(-c(lunaid, whoqol_avg_score_z)))
y <- as.matrix(dat %>% select(c(whoqol_avg_score_z)))

# predictors for which there were significant age effects (when controlling for FDR)
    in GAMMs
X_sig <- as.matrix(dat %>%
                     select(-c(lunaid, whoqol_avg_score_z)) %>%
                     select(c(anti_perc_correct_nodrop_rint,
                              anti_perc_correct_nodrop_rslope,
                              anti_perc_error_corrected_nodrop_rint,
                              anti_perc_error_corrected_rslope,
                              anti_avg_lat_correct_trials_rint,
                              anti_avg_lat_correct_trials_rslope,
                              L_FEF_anti_corr_v_baseline_rint,
                              L_FEF_anti_corr_v_baseline_rslope,
                              L_pPC_anti_corr_v_baseline_rint,
                              L_pPC_anti_corr_v_baseline_rslope,
```

```
                                R_pPC_anti_corr_v_baseline_rint,
                                R_pPC_anti_corr_v_baseline_rslope,
                                R_dlPFC_anti_corr_v_baseline_rint,
                                R_dlPFC_anti_corr_v_baseline_rslope,
                                dACC_anti_errcorr_v_baseline_rint,
                                dACC_anti_errcorr_v_baseline_rslope)))

## fit the inital elastic net model
# sequence of alpha parameter values to try
# does not include 0 or 1 because those values correspond to ridge and lasso
alphas <- seq(from = 0.1, to = 0.9, length.out = 17)

# create k = 10 folds for cross-validation
# explicitly setting folds to allow for reproducibility
set.seed(100)
folds <- createFolds(y = y, k = 10, list = FALSE, returnTrain = FALSE)

# simultaneous tuning of alpha and lambda
fit <- ensr(x = X_sig,
            y = y,
            alphas = alphas,
            foldid = folds,
            standardize = FALSE)

# alpha and lambda values that minimized CV-MSE
summary(fit)[cvm == min(cvm)]
summary(fit)[cvm == min(cvm)]$alpha
summary(fit)[cvm == min(cvm)]$lambda

# fit the model using the alpha and lambda values that minimized CV-MSE
glmnetfit <- glmnet(x = X_sig,
                    y = y,
                    alpha = summary(fit)[cvm == min(cvm)]$alpha,
                    lambda = summary(fit)[cvm == min(cvm)]$lambda,
                    family = 'gaussian',
                    standardize = FALSE)

coef(glmnetfit) # estimated beta coefficients
glmnetfit$dev.ratio # model R^2


# organize coefficients into dataframe
coefs_glmnet_fit <- as.data.frame(as.matrix(coef(glmnetfit)))
colnames(coefs_glmnet_fit) <- 'beta_hat'
coefs_glmnet_fit <- tibble::rownames_to_column(coefs_glmnet_fit, 'predictor')
coefs_glmnet_fit


## use bootstrap-enhanced procedure to:
## (1) derive confidence intervals and
## (2) calculate variable inclusion probabilities
# number of bootstrap samples and subjects
B <- 5000
subjs <- 1:nrow(data_for_boot)

coefs_boot <- data.frame(Predictor = predictors, stringsAsFactors = FALSE)

# perform bootstrap procedure
for (i in 1:B){
```

```
  # resample the data with replacement
  set.seed(i*10)
  boot_sample <- sample(subjs, size = 50, replace = TRUE)
  y_boot <- y[boot_sample]
  X_sig_boot <- X_sig[boot_sample, 1:ncol(X_sig)]

  # fit the elastic net model on the bootstrap sample
  glmnet_boot <- glmnet(x = X_sig_boot,
                        y = y_boot,
                        alpha = summary(fit)[cvm == min(cvm)]$alpha,
                        lambda = summary(fit)[cvm == min(cvm)]$lambda,
                        family = 'gaussian',
                        standardize = FALSE)

  # store the estimated coefficients
  coefs <- as.data.frame(as.matrix(coef(glmnet_boot))) %>%
    tibble::rownames_to_column(var = 'Predictor')
  colnames(coefs_nocv)[2] <- paste0('Beta_boot_', i)

  coefs_boot <- full_join(coefs_boot, coefs, by = 'Predictor')
}

coefs_boot

# transpose bootstrap coefficient dataframe from wide to long format
coefs_boot_long_colnames <- coefs_boot_nocv[,1]
coefs_boot_long <- as.data.frame(t(coefs_boot[,-1]), stringsAsFactors = FALSE)
colnames(coefs_boot_long) <- coefs_boot_long_colnames
coefs_boot_long

# 95% bootstrapped CIs
coefs_quantiles <- as.data.frame(t(sapply(coefs_boot_long,
                                          quantile,
                                          probs = c(0.025, 0.975),
                                          names = TRUE)))
coefs_quantiles <- tibble::rownames_to_column(coefs_quantiles, 'predictor')
coefs_quantiles

# variable inclusion probabilities
VIP <- as.data.frame(cbind(coefs_boot[,1],
                           rowSums(coefs_boot[,-1] != 0)),
                     stringsAsFactors = FALSE)
colnames(VIP) <- c('predictor', 'times_selected')
VIP$times_selected <- as.numeric(VIP$times_selected)
VIP$prop_selected <- VIP$times_selected/B
VIP

# join the results
coefs_and_CI <- full_join(coefs_glmnet_fit, coefs_quantiles)
coefs_and_CI_and_VIP <- full_join(coefs_and_CI, VIP) %>%
  select(-c('times_selected')) %>%
  rename(VIP = prop_selected)
coefs_and_CI_and_VIP

# save the results
write.csv(coefs_and_CI_and_VIP, 'Data/bootstrap_enhanced_enet_results_20200329.csv')


## create LaTeX table to include in document
```

```r
# clean up predictor names
coefs_and_CI_and_VIP$predictor <- gsub(pattern = '_',
                                       replacement = ' ',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'anti corr v baseline',
                                       replacement = 'percent signal change correct trials',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'anti errcorr v baseline',
                                       replacement = 'percent signal change error-corrected trials',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'rint',
                                       replacement = 'intercept',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'rslope',
                                       replacement = 'slope',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'anti perc',
                                       replacement = 'Antisaccade proportion',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'error corrected',
                                       replacement = 'error-corrected trials',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'nodrop',
                                       replacement = '',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'anti avg lat',
                                       replacement = 'Antisaccade average latency',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = '  ',
                                       replacement = ' ',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'Antisaccade proportion correct',
                                       replacement = 'Antisaccade proportion correct trials',
                                       x = coefs_and_CI_and_VIP$predictor)
coefs_and_CI_and_VIP$predictor <- gsub(pattern = 'percent signal change',
                                       replacement = 'activation',
                                       x = coefs_and_CI_and_VIP$predictor)

# save latex table code
print(xtable(coefs_and_CI_and_VIP,
             digits = c(0,0,4,4,4,4),
             type = 'latex'),
      file = 'bootstrap_enhanced_enet_20200329.tex')
```

```r
##########################################
### Post-hoc analysis for elastic net ###
##########################################

setwd('/Users/jenniferfedor/Documents/Biostats MS/Spring 2020/Thesis')

library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(gridExtra)
```

```r
library(mgcv)

## data for gamms
dat_gamm <- read.csv('Data/final_eye_scan_data_for_analysis_20200228.csv',
                     stringsAsFactors = FALSE)

dat_gamm <- dat_gamm %>%
  select(-c('X')) %>%
  mutate(lunaid = as.factor(lunaid)) %>%
  mutate(date = as.Date(date, format = '%Y-%m-%d')) %>%
  mutate(dob = as.Date(dob, format = '%Y-%m-%d')) %>%
  mutate(sex = as.factor(sex))

dat_gamm <- within(dat_gamm,
                   level_edu_mother_cat <- relevel(as.factor(level_edu_mother_cat),
                                                   ref = 'Completed high school'))
dat_gamm$age_c <- dat$age - mean(dat$age)


## whoqol data
dat_whoqol <- read.csv('Data/data_for_elastic_net_20200317.csv',
                       stringsAsFactors = FALSE)

# median whoqol composite score
med_score <- median(dat_whoqol$whoqol_avg_score_z)
# median split to create high and low QOL groups
dat_whoqol <- dat_whoqol %>%
  select(c('lunaid', 'whoqol_avg_score_z')) %>%
  mutate(whoqol_group = case_when(whoqol_avg_score_z < med_score ~ 'low',
                                  whoqol_avg_score_z >= med_score ~ 'high'))

write.csv(dat_whoqol, 'Data/whoqol_group_labels_20200414.csv')


## merge data for gamms and whoqol groups
dat_plus_whoqol_grp <- merge(dat_gamm, dat_whoqol)
levels(dat_plus_whoqol_grp$whoqol_group) <- c('High', 'Low')
# ordered factor
dat_plus_whoqol_grp$ordered_whoqol_group <- ordered(dat_plus_whoqol_grp$whoqol_group,
                                                    levels = c('High', 'Low'))


## refit GAMMS for three predictors that had high VIPs with smooth age x QOL group interaction and plot two group trajectories
# 1. Proportion error corrected antisaccade trials
model_formula <- as.formula("anti_perc_error_corrected_nodrop ~ ordered_whoqol_group +
                            s(age_c, k = 10, fx = FALSE, bs = 'tp') +
                            s(age_c, by = ordered_whoqol_group, k = 10, fx = FALSE, bs = 'tp')")

gamm_anti_perc_error_corrected_nodrop_bywhoqol <- gamm(as.formula(model_formula),
                                                       family = 'quasibinomial',
                                                       random = list(lunaid=~age_c),
                                                       data = dat_plus_whoqol_grp,
                                                       method = 'REML')

summary(gamm_anti_perc_error_corrected_nodrop_bywhoqol$gam)
```

```r

# 2. L FEF activation
model_formula <- as.formula("L_FEF_anti_corr_v_baseline ~ ordered_whoqol_group +
                            s(age_c, k = 10, fx = FALSE, bs = 'tp') +
                           s(age_c, by = ordered_whoqol_group, k = 10, fx = FALSE,
                              bs = 'tp')")

gamm_L_FEF_anti_corr_v_baseline_bywhoqol <- gamm(as.formula(model_formula),
                                                 random = list(lunaid=~age_c),
                                                 data = dat_plus_whoqol_grp,
                                                 method = 'REML')

summary(gamm_L_FEF_anti_corr_v_baseline_bywhoqol$gam)


# 3. R dlPFC activation
model_formula <- as.formula("R_dlPFC_anti_corr_v_baseline ~ ordered_whoqol_group +
                            s(age_c, k = 10, fx = FALSE, bs = 'tp') +
                            s(age_c, by = ordered_whoqol_group, k = 10, fx = FALSE,
                                bs = 'tp')")

gamm_R_dlPFC_anti_corr_v_baseline_bywhoqol <- gamm(as.formula(model_formula),
                                                   random = list(lunaid=~age_c),
                                                   data = dat_plus_whoqol_grp,
                                                   method = 'REML')

summary(gamm_R_dlPFC_anti_corr_v_baseline_bywhoqol$gam)


## function to plot groups' developmental trajectories
plot_gamm_int <- function(model, int_var, smooth_var, y_axis_label = ''){
  mod <- get(model)$gam
  s <- summary(mod)
  df <- mod$model
  df <- df %>% mutate(age = (age = age_c + mean(dat_gamm$age)))

  response <- as.character(mod$terms[[2]])
  labs <- levels(df[, int_var]) # groups for interaction term

  # new data to predict for
  np <- 10000 # number of values
  newdat <- data.frame(matrix(data = NA, nrow = np, ncol = 0))
  newdat[, smooth_var] <- seq(min(df[, smooth_var], na.rm = TRUE),
                              max(df[, smooth_var], na.rm = TRUE),
                              length.out = np)
  newdat <- do.call('rbind',
            replicate(length(labs),
                      newdat,
                      simplify = FALSE))

  newdat[, int_var] <- ordered(rep(labs, each = np), levels = labs)

  # fitted values and standard errors for pointwise CIs
  pred <- data.frame(predict.gam(mod,
                                 newdata = newdat,
                                 type = 'response',
                                 se.fit = TRUE))
  pred <- cbind(newdat, pred)
```

```
  pred[, response] <- 1
  pred$age <- pred$age_c + mean(dat$age)
  pred$ci_low <- pred$fit - 1.96*pred$se.fit
  pred$ci_high <- pred$fit + 1.96*pred$se.fit

  # plot trajectories for each group
  p <- ggplot() +
    geom_ribbon(data = pred,
                aes_string(x = 'age',
                           ymin = 'ci_low',
                           ymax = 'ci_high',
                           fill = int_var),
                alpha = .5, linetype = 0) +
    scale_fill_manual(values = c('#D1474B', '#476FD1')) +
    geom_line(data = pred,
              aes_string(x = 'age',
                  y = 'fit',
                  color = int_var), size = 1) +
    scale_color_manual(values = c('#D1474B', '#476FD1')) +
    labs(x = 'Age (years)',
         y = y_axis_label,
         fill = 'QOL group',
         color = 'QOL group') +
    theme_bw() +
    theme(panel.border = element_blank(),
          panel.grid.major = element_blank(),
          #legend.position = 'none',
          panel.grid.minor = element_blank(),
          axis.line = element_line(colour = "black")) +
    theme(axis.title = element_text(size = 11),
          axis.text = element_text(size = 10),
          legend.title=element_text(size = 11),
          legend.text = element_text(size = 10))
  return(p)
}


## plots of group trajectories for the three models
p1 <- plot_gamm_int(model = 'gamm_anti_perc_error_corrected_nodrop_bywhoqol',
                    int_var = 'ordered_whoqol_group',
                    smooth_var = 'age_c',
                    y_axis_label = 'Proportion error-corrected trials')
p1 <- p1 + theme(legend.position = 'none')
p1

p2 <- plot_gamm_int(model = 'gamm_L_FEF_anti_corr_v_baseline_bywhoqol',
                    int_var = 'ordered_whoqol_group',
                    smooth_var = 'age_c',
                    y_axis_label = 'L FEF percent signal change')
p2 <- p2 + theme(legend.position = 'none')
p2

p3 <- plot_gamm_int(model = 'gamm_R_dlPFC_anti_corr_v_baseline_bywhoqol',
                    int_var = 'ordered_whoqol_group',
                    smooth_var = 'age_c',
                    y_axis_label = 'R dlPFC percent signal change')
p3

# arrange plots
```

```r
plots_by_whoqol_group <- gridExtra::grid.arrange(
  p1, p2, grid::nullGrob(), p3, grid::nullGrob(),
  layout_matrix = matrix(c(1,1,1,2,2,2,
                           3,4,4,4,4,5),
                         byrow = TRUE,
                         ncol = 6))
plots_by_whoqol_group

ggsave('Plots/plots_by_whoqol_group.png',
       plots_by_whoqol_group,
       width = 10, height = 8, units = 'in')
```

Bibliography

1. Huttenlocher, P. R. & Dabholkar, A. S. Regional differences in synaptogenesis in human cerebral cortex. *The Journal of Comparative Neurology* **387,** 167–178. ISSN: 0021-9967 (Oct. 20, 1997).
2. Luna, B. *et al.* Maturation of widely distributed brain function subserves cognitive development. *NeuroImage* **13,** 786–793. ISSN: 1053-8119 (May 2001).
3. Luna, B. Developmental changes in cognitive control through adolescence. *Advances in Child Development and Behavior* **37,** 233–278. ISSN: 0065-2407 (2009).
4. Diamond, A. Executive Functions. *Annual review of psychology* **64,** 135–168. ISSN: 0066-4308 (2013).
5. Ochsner, K. & Gross, J. The cognitive control of emotion. *Trends in Cognitive Sciences* **9,** 242–249. ISSN: 13646613 (May 2005).
6. Williams, P. G., Tinajero, R. & Suchy, Y. *Executive Functioning and Health* (Oxford University Press, Nov. 6, 2017).
7. Luna, B., Paulsen, D. J., Padmanabhan, A. & Geier, C. The Teenage Brain: Cognitive Control and Motivation. *Current Directions in Psychological Science* **22,** 94–100. ISSN: 0963-7214 (Apr. 1, 2013).
8. Cunningham, R. M., Walton, M. A. & Carter, P. M. The Major Causes of Death in Children and Adolescents in the United States. *The New England journal of medicine* **379,** 2468–2475. ISSN: 0028-4793 (Dec. 20, 2018).
9. Centers for Disease Control and Prevention (CDC) National Center for Health Statistics. *Adolescent Health* `https://www.cdc.gov/nchs/fastats/adolescent-health.htm` (2020).
10. Arnsten, A. F. T. & Rubia, K. Neurobiological Circuits Regulating Attention, Cognitive Control, Motivation, and Emotion: Disruptions in Neurodevelopmental Psychiatric Disorders. *Journal of the American Academy of Child & Adolescent Psychiatry* **51,** 356–367. ISSN: 0890-8567 (Apr. 1, 2012).
11. World Health Organization (WHO). *Adolescent health epidemiology* WHO. `https://www.who.int/maternal_child_adolescent/epidemiology/adolescence/en/` (2020).

12. Healthy People 2020. *Adolescent Health* U.S. Department of Health and Human Services, Office of Disease Prevention and Health Promotion. `https://www.healthypeople.gov/2020/topics-objectives/topic/Adolescent-Health` (2020).

13. Lenartowicz, A., Kalar, D. J., Congdon, E. & Poldrack, R. A. Towards an Ontology of Cognitive Control. *Topics in Cognitive Science* **2,** 678–692. ISSN: 1756-8765 (2010).

14. Miller, E. K. & Cohen, J. D. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience* **24,** 167–202. ISSN: 0147-006X (2001).

15. Miyake, A. *et al.* The unity and diversity of executive functions and their contributions to complex "Frontal Lobe" tasks: a latent variable analysis. *Cognitive Psychology* **41,** 49–100. ISSN: 0010-0285 (Aug. 2000).

16. Diamond, A. Developmental time course in human infants and infant monkeys, and the neural bases of, inhibitory control in reaching. *Annals of the New York Academy of Sciences* **608,** 637–669. ISSN: 0077-8923 (1990).

17. Diamond, A. & Goldman-Rakic, P. S. Comparison of human infants and rhesus monkeys on Piaget's AB task: evidence for dependence on dorsolateral prefrontal cortex. *Experimental Brain Research* **74,** 24–40. ISSN: 0014-4819 (1989).

18. Moilanen, K. L., Shaw, D. S., Dishion, T. J., Gardner, F. & Wilson, M. Predictors of Longitudinal Growth in Inhibitory Control in Early Childhood. *Social Development* **19,** 326–347. ISSN: 1467-9507 (2010).

19. Munoz, D. P., Broughton, J. R., Goldring, J. E. & Armstrong, I. T. Age-related performance of human subjects on saccadic eye movement tasks. *Experimental Brain Research* **121,** 391–400. ISSN: 0014-4819 (Aug. 1998).

20. Luna, B., Garver, K. E., Urban, T. A., Lazar, N. A. & Sweeney, J. A. Maturation of cognitive processes from late childhood to adulthood. *Child Development* **75,** 1357–1372. ISSN: 0009-3920 (Oct. 2004).

21. Hallett, P. E. Primary and secondary saccades to goals defined by instructions. *Vision Research* **18,** 1279–1296. ISSN: 0042-6989 (Jan. 1, 1978).

22. Ordaz, S. J., Foran, W., Velanova, K. & Luna, B. Longitudinal Growth Curves of Brain Function Underlying Inhibitory Control through Adolescence. *The Journal of Neuroscience* **33,** 18109–18124. ISSN: 0270-6474 (Nov. 13, 2013).

23. Glover, G. H. Overview of Functional Magnetic Resonance Imaging. *Neurosurgery clinics of North America* **22,** 133–139. ISSN: 1042-3680 (Apr. 2011).

24. Fuster, J. *The Prefrontal Cortex* 3rd ed. 461 pp. ISBN: 978-0-12-408060-7 (Raven Press, New York, 1997).

25. Gogtay, N. *et al.* Dynamic mapping of human cortical development during childhood through early adulthood. *Proceedings of the National Academy of Sciences of the United States of America* **101,** 8174–8179. ISSN: 0027-8424 (May 25, 2004).

26. Tamm, L., Menon, V. & Reiss, A. L. Maturation of brain function associated with response inhibition. *Journal of the American Academy of Child and Adolescent Psychiatry* **41,** 1231–1238. ISSN: 0890-8567 (Oct. 2002).

27. Rubia, K. *et al.* Functional frontalisation with age: mapping neurodevelopmental trajectories with fMRI. *Neuroscience and Biobehavioral Reviews* **24,** 13–19. ISSN: 0149-7634 (Jan. 2000).

28. Casey, B. J. *et al.* A Developmental Functional MRI Study of Prefrontal Activation during Performance of a Go-No-Go Task. *Journal of Cognitive Neuroscience* **9,** 835–847. ISSN: 0898-929X (Nov. 1997).

29. Durston, S. *et al.* A neural basis for the development of inhibitory control. *Developmental Science* **5,** F9–F16. ISSN: 1467-7687. `https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-7687.00235` (2020) (2002).

30. Luna, B., Padmanabhan, A. & O'Hearn, K. What has fMRI told us about the Development of Cognitive Control through Adolescence? *Brain and cognition* **72,** 101. ISSN: 0278-2626 (Feb. 2010).

31. Luna, B., Marek, S., Larsen, B., Tervo-Clemmens, B. & Chahal, R. An Integrative Model of the Maturation of Cognitive Control. *Annual review of neuroscience* **38,** 151–170. ISSN: 0147-006X (July 8, 2015).

32. Ghuman, A. S., Bar, M., Dobbins, I. G. & Schnyer, D. M. The effects of priming on frontal-temporal communication. *Proceedings of the National Academy of Sciences of the United States of America* **105,** 8405–8409. ISSN: 1091-6490 (June 17, 2008).

33. Sweeney, J. A. *et al.* Positron emission tomography study of voluntary saccadic eye movements and spatial working memory. *Journal of Neurophysiology* **75,** 454–468. ISSN: 0022-3077 (Jan. 1996).

34. Velanova, K., Wheeler, M. E. & Luna, B. Maturational Changes in Anterior Cingulate and Frontoparietal Recruitment Support the Development of Error Processing and Inhibitory Control. *Cerebral Cortex* **18,** 2505–2522. ISSN: 1047-3211 (Nov. 2008).

35. Carter, C. S. *et al.* Anterior cingulate cortex, error detection, and the online monitoring of performance. *Science* **280,** 747–749. ISSN: 0036-8075 (May 1, 1998).

36. Hayes, R. D. & Reeve, B. B. in *Epidemiology and demography in public health* (eds Killewo, J., Heggenhougen, H. K. & Quah, S. R.) 195–205 (Academic Press, San Diego, 2010).

37. The Whoqol Group. The World Health Organization Quality of Life assessment (WHOQOL): position paper from the World Health Organization. *Social Science & Medicine* **41,** 1403–1409. ISSN: 0277-9536 (Nov. 1995).

38. Cherepanov, D. *et al.* Gender Differences in Multiple Underlying Dimensions of Health-Related Quality of Life are Associated with Sociodemographic and Socioeconomic Status. *Medical care* **49,** 1021–1030. ISSN: 0025-7079 (Nov. 2011).

39. Meade, T. & Dowswell, E. Adolescents' health-related quality of life (HRQoL) changes over time: a three year longitudinal study. *Health and Quality of Life Outcomes* **14.** ISSN: 1477-7525 (Jan. 25, 2016).

40. Kim, J.-H. & Park, E.-C. Impact of socioeconomic status and subjective social class on overall and health-related quality of life. *BMC Public Health* **15,** 1–15. ISSN: 1471-2458 (Dec. 2015).

41. Wang, W.-C., Yao, G., Tsai, Y.-J., Wang, J.-D. & Hsieh, C.-L. Validating, improving reliability, and estimating correlation of the four subscales in the WHOQOL-BREF using multidimensional Rasch analysis. *Quality of Life Research: An International Journal of Quality of Life Aspects of Treatment, Care and Rehabilitation* **15,** 607–620. ISSN: 0962-9343 (May 2006).

42. Diener, E. & Ryan, K. Subjective Well-Being: A General Overview. *South African Journal of Psychology* **39,** 391–406. ISSN: 0081-2463 (Dec. 1, 2009).

43. Jefferson, A. L., Paul, R. H., Ozonoff, A. & Cohen, R. A. Evaluating elements of executive functioning as predictors of instrumental activities of daily living (IADLs). *Archives of clinical neuropsychology : the official journal of the National Academy of Neuropsychologists* **21,** 311–320. ISSN: 0887-6177 (May 2006).

44. Davis, J. C., Marra, C. A., Najafzadeh, M. & Liu-Ambrose, T. The independent contribution of executive functions to health related quality of life in older women. *BMC Geriatrics* **10,** 16. ISSN: 1471-2318 (Apr. 1, 2010).

45. Enticott, P. G., Ogloff, J. R. P. & Bradshaw, J. L. Associations between laboratory measures of executive inhibitory control and self-reported impulsivity. *Personality and Individual Differences* **41,** 285–294. ISSN: 0191-8869 (July 1, 2006).

46. Chamberlain, S. R. & Grant, J. E. Relationship between quality of life in young adults and impulsivity/compulsivity. *Psychiatry Research* **271,** 253–258. ISSN: 0165-1781 (Jan. 2019).

47. Nelder, J. A. & Wedderburn, R. W. M. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)* **135,** 370. ISSN: 00359238 (1972).

48. Fox, J. *Applied Regression Analysis and Generalized Linear Models* 817 pp. ISBN: 978-1-4833-1088-6 (SAGE Publications, Mar. 18, 2015).

49. *Regression methods in biostatistics: linear, logistic, survival, and repeated measures models* (ed Vittinghoff, E.) (Springer, New York, 2005). 340 pp. ISBN: 978-0-387-20275-4.

50. Hawkins, D. M. The Problem of Overfitting. *Journal of chemical information and computer sciences* **44,** 1–12. ISSN: 0095-2338 (Jan. 1, 2004).

51. Tjørve, K. M. C. & Tjørve, E. The use of Gompertz models in growth analyses, and new Gompertz-model approach: An addition to the Unified-Richards family. *PLOS ONE* **12,** e0178691. ISSN: 1932-6203 (June 5, 2017).

52. Hastie, T. & Tibshirani, R. Generalized Additive Models. *Statistical Science* **1,** 297–310. ISSN: 0883-4237, 2168-8745 (Aug. 1986).

53. Wood, S. N. *Generalized Additive Models: an introduction with R* (Chapman & Hall/CRC, New York, 2006).

54. Lin, X. & Zhang, D. Inference in Generalized Additive Mixed Models by Using Smoothing Splines. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61,** 381–400. ISSN: 1369-7412 (1999).

55. Marra, G. & Radice, R. Penalised regression splines: theory and application to medical research. *Statistical Methods in Medical Research* **19,** 107–125. ISSN: 0962-2802 (Apr. 1, 2010).

56. Wood, S. N. Thin plate regression splines. *Journal of the Royal Statistical Society, Series B* **65,** 95–114 (2003).

57. Wood, S. *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation* R package version 1.8-28. 2019. `https://CRAN.R-project.org/package=mgcv`.

58. Hedeker, D. & Gibbons, R. D. *Longitudinal Data Analysis* 369 pp. ISBN: 978-0-470-03647-1 (John Wiley & Sons, Hoboken, NJ, May 12, 2006).

59. Chen, C. Generalized additive mixed models. *Communications in Statistics - Theory and Methods* **29,** 1257–1271. ISSN: 0361-0926 (Jan. 1, 2000).

60. Mullah, M. A. S., Hanley, J. A. & Benedetti, A. Modeling perinatal mortality in twins via generalized additive mixed models: a comparison of estimation approaches. *BMC Medical Research Methodology* **19.** ISSN: 1471-2288 (Nov. 15, 2019).

61. Sóskuthy, M. Generalised additive mixed models for dynamic analysis in linguistics: a practical introduction. *arXiv:1703.05339 [stat].* `http://arxiv.org/abs/1703.05339` (Mar. 15, 2017).

62. Alexander-Bloch, A. F. *et al.* Abnormal Cortical Growth in Schizophrenia Targets Normative Modules of Synchronized Development. *Biological Psychiatry. Brain Development and Connectivity in Schizophrenia* **76,** 438–446. ISSN: 0006-3223 (Sept. 15, 2014).

63. Forde, N. J. *et al.* Healthy cortical development through adolescence and early adulthood. *Brain Structure & Function* **222,** 3653–3663. ISSN: 1863-2661 (Nov. 2017).

64. Tamnes, C. K. *et al.* Development of the Cerebral Cortex across Adolescence: A Multisample Study of Inter-Related Longitudinal Changes in Cortical Volume, Surface Area, and Thickness. *The Journal of Neuroscience* **37,** 3402–3412. ISSN: 0270-6474 (Mar. 22, 2017).

65. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations* 354 pp. ISBN: 978-1-4987-1217-0 (CRC Press, May 7, 2015).

66. *An introduction to statistical learning: with applications in R* (eds James, G., Witten, D., Hastie, T. & Tibshirani, R.) *Springer texts in statistics* **103** (Springer, New York, 2013). 426 pp. ISBN: 978-1-4614-7137-0.

67. Mumford, J. A. & Nichols, T. Power Calculation for Group fMRI Studies Accounting for Arbitrary Design and Temporal Autocorrelation. *NeuroImage* **39,** 261–268. ISSN: 1053-8119 (Jan. 1, 2008).

68. Yeung, A. W. K. An Updated Survey on Statistical Thresholding and Sample Size of fMRI Studies. *Frontiers in Human Neuroscience* **12.** ISSN: 1662-5161 (2018).

69. Hoerl, A. E. & Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **12,** 55–67. ISSN: 0040-1706. `https://amstat.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634` (2020) (Feb. 1, 1970).

70. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58,** 267–288. ISSN: 2517-6161 (1996).

71. Zou, H. & Hastie, T. Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B* **67,** 301–320 (2005).

72. Waldmann, P., Mészáros, G., Gredler, B., Fürst, C. & Sölkner, J. Evaluation of the lasso and the elastic net in genome-wide association studies. *Frontiers in Genetics* **4.** ISSN: 1664-8021 (2013).

73. Ryali, S., Chen, T., Supekar, K. & Menon, V. Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage* **59,** 3852–3861. ISSN: 1095-9572 (Feb. 15, 2012).

74. Weschler, D. *Weschler Adult Intelligence Scale - Revised* (The Psychological Corporation, San Antonio, TX, 1999).

75. Gitelman, D. R. ILAB: a program for postexperimental eye movement analysis. *Behavior Research Methods, Instruments, & Computers: A Journal of the Psychonomic Society, Inc* **34,** 605–612. ISSN: 0743-3808 (Nov. 2002).

76. Cox, R. W. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research* **29,** 162–173 (1996).

77. Yarkoni, T., Poldrack, R. A., Nichols, T. E., Van Essen, D. C. & Wager, T. D. Large-scale automated synthesis of human functional neuroimaging data. *Nature Methods* **8,** 665–670. ISSN: 1548-7091, 1548-7105 (Aug. 2011).

78. Cosgrove, K. P., Mazure, C. M. & Staley, J. K. Evolving Knowledge of Sex Differences in Brain Structure, Function, and Chemistry. *Biological Psychiatry* **62,** 847–855. ISSN: 0006-3223, 1873-2402 (Oct. 15, 2007).

79. Jackson, M., Kiernan, K. & McLanahan, S. Maternal Education, Changing Family Circumstances, and Children's Skill Development in the United States and UK. *The Annals of the American Academy of Political and Social Science* **674,** 59–84. ISSN: 0002-7162 (Nov. 2017).

80. Harding, J. F., Morris, P. A. & Hughes, D. The Relationship Between Maternal Education and Children's Academic Outcomes: A Theoretical Framework. *Journal of Marriage and Family* **77,** 60–76. ISSN: 1741-3737 (2015).

81. Sullivan, K. J., Shadish, W. R. & Steiner, P. M. An introduction to modeling longitudinal data with generalized additive models: Applications to single-case designs. *Psychological Methods* **20,** 26–42. ISSN: 1082-989X (2015).

82. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)* **57,** 289–300. ISSN: 0035-9246 (1995).

83. Simpson, G. *gratia: Graceful 'ggplot'-Based Graphics and Other Functions for GAMs Fitted Using 'mgcv'* R package version 0.3.0. 2020. `https://CRAN.R-project.org/package=gratia`.

84. Lockhart, R., Taylor, J., Tibshirani, R. J. & Tibshirani, R. A significance test for the lasso. *The Annals of Statistics* **42,** 413–468. ISSN: 0090-5364 (Apr. 2014).

85. Bunea, F. *et al.* Penalized Least Squares Regression Methods and Applications to Neuroimaging. *NeuroImage* **55,** 1519–1527. ISSN: 1053-8119 (Apr. 15, 2011).

86. Abram, S. V. *et al.* Bootstrap Enhanced Penalized Regression for Variable Selection with Neuroimaging Data. *Frontiers in Neuroscience* **10.** ISSN: 1662-4548 (July 28, 2016).

87. DeWitt, P. *ensr: Elastic Net SearcheR* R package version 0.1.0. 2019. `https://CRAN.R-project.org/package=ensr`.

88. Friedman, J. *et al. glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models* R package version 3.0-2. 2019. `https://CRAN.R-project.org/package=glmnet`.

89. Hwang, K., Velanova, K. & Luna, B. Strengthening of Top-Down Frontal Cognitive Control Networks Underlying the Development of Inhibitory Control: A Functional Magnetic Resonance Imaging Effective Connectivity Study. *The Journal of Neuroscience* **30,** 15535–15545. ISSN: 0270-6474 (Nov. 17, 2010).

90. Button, K. S. *et al.* Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience* **14,** 365–376. ISSN: 1471-003X, 1471-0048 (May 2013).

91. Thomas, C. G., Harshman, R. A. & Menon, R. S. Noise reduction in BOLD-based fMRI using component analysis. *NeuroImage* **17,** 1521–1537. ISSN: 1053-8119 (Nov. 2002).